Context-enhanced detection of electrophysiology catheters in noisy fluoroscopy images

Franken, E.M.

Award date:
2004

Eindhoven University of Technology
Department of Electrical Engineering
Control Systems

CONFIDENTIAL

Context-Enhanced Detection of Electrophysiology Catheters in Noisy Fluoroscopy Images

by E.M. Franken

Master of Science thesis

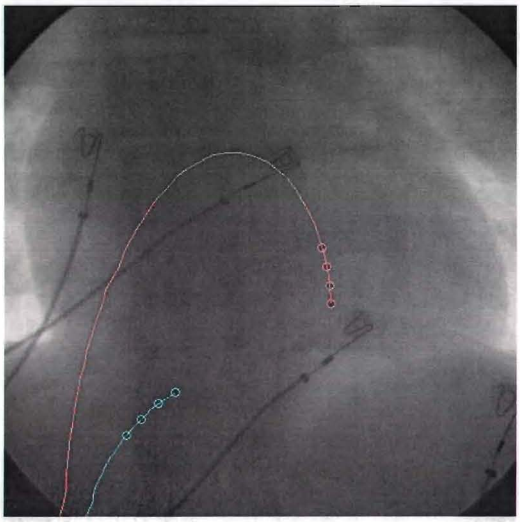
Project period: October 2003 – September 2004
Report Number: 04A/08

Supervisors:
Dipl.phys. M.A. van Almsick (Dep. of Biomedical Engineering, TU/e)
Dr.ir. P.M.J. Rongen (Philips Medical Systems Best)

Committee members:
Prof.dr.ir. P.P.J. van den Bosch (Dep. of Electrical Engineering, TU/e)
Prof.dr.ir. B.M. ter Haar Romeny (Dep. of Biomedical Engineering, TU/e)

Context-Enhanced Detection of Electrophysiology Catheters in Noisy Fluoroscopy Images

by Erik Franken



Master's thesis

September 2004

CONFIDENTIAL

Commissioned by:
Prof. dr. ir. P.P.J. van den Bosch (Dep. of Electrical Engineering, TU/e)
Prof. dr. ir. B.M. ter Haar Romeny (Dep. of Biomedical Engineering, TU/e)
Dipl.ing. M.A. van Almsick (Dep. of Biomedical Engineering, TU/e)
Dr. ir. P.M.J. Rongen (X-ray Predevelopment, Philips Medical Systems Best)

Abstract

Cardiac catheter ablation is a minimally invasive medical procedure to treat patients with heart rhythm disorders. Electrophysiology (EP) catheters are inserted through small skin incisions and are threaded through the blood vessels to the heart. The EP catheters contain a number of electrodes used for recording intracardiac electrograms and ablating problematic spots of the myocardium. For the navigation of the catheters through the body, X-ray fluoroscopy imaging is necessary, resulting in a high cumulated radiation dose during lengthy ablation procedures. A reduction of the X-ray dose leads to noisy images. Therefore, there is a demand for image analysis techniques that can extract catheters from very noisy images. Information on the catheter position can be helpful for image enhancement and automatic dose control.

In this work, we investigate the use of spatial context to extract the EP catheters with higher certainty. Inspired by the biological visual system we use context filters, which enhance elongated structures in an image. First we find locally line-like structures and blob-like structures in the image using steerable filters. The resulting local feature data is the input for the context enhancement step. We use an adapted version of the tensor voting technique by Medioni et al., in which an "8-shaped" voting field enables communication of local feature information encoded in a tensor field. We propose to use a steerable voting field. This enables us to derive a computational scheme of tensor voting, for which the tensor broadcast operation can be rewritten as a set of complex-valued convolutions, leading to a more efficient implementation.

After context enhancement, we extract the EP catheters from the resulting feature images. We use a multi-step approach, in which the most promising structures in the image are grouped to the final objects of interest, i.e. the EP catheters. For this purpose, application-specific grouping rules are defined, which use knowledge of the EP catheters concerning electrode spacing, curvature behavior, etcetera.

To prove the potential of the proposed methods, we compare the performance with and without the use of tensor voting. We conclude that the use of tensor voting leads to an improvement in feature image quality, and an increase in the success rate of EP catheter extraction. Especially for clinical EP catheter images with additional noise added, we observe a large increase in success rate with tensor voting.

Preface

This thesis describes the work I performed for my master's project in Information Technology, between October 2003 and September 2004. The work, entitled "Context-Enhanced Detection of Electrophysiology Catheters in Noisy Fluoroscopy Images", was performed half-time at the Biomedical Image Analysis group of the Department of Biomedical Technology of the Technische Universiteit Eindhoven, and half-time at X-ray predevelopment, Philips Medical System, in Best. Sometimes, this close collaboration between university and company was hard. The approach and goals of the two partners are quite different. I had to find my way somewhere in between the theoretical approach and the more practical and pragmatic approach. Although difficult, this also made it very interesting. I learned a lot about the differences between companies and universities. I also learned to collaborate with people from a lot of different disciplines: electrical engineers, physicists, mathematicians, etcetera.

I would like to thank a lot of people involved in this project. From the department of Electrical Engineering, I thank prof. Paul van den Bosch. From the group Biomedical Image Analysis I would like to thank prof. Bart ter Haar Romeny, Markus van Almsick, Remco Duits, Luc Florack, and all other group members. From Philips Medical Systems I would like to thank Peter Rongen, Herman Stegehuis, Geert Gijsbers, Erik Hummel, Niels Nijhof, Peter van den Houten, all other colleagues, and my colleague students for being nice roommates. I also thank my family and friends for being supportive during this period.

Notations and Symbols

Typesetting conventions

$\mathbf{v}$	Vector
$\bar{a}$	Complex conjugate of complex number a
$f(x,y)$	Scalar function expressed in cartesian coordinate system
$\tilde{f}(r,\varphi)$	Scalar function expressed in polar coordinate system
M	Matrix, or tensor (Subsection 4.2.1)
T'	Tensor written as a vector (Subsection 4.2.1)
$\mathrm{A}(x,y)$	Tensor field expressed in cartesian coordinate system
$\tilde{\mathrm{A}}(r,\varphi)$	Tensor field expressed in polar coordinate system
$\mathcal{P}$	Set (no ordering) or list (with ordering)
$\check{\mathcal{P}}$	List $\mathcal{P}$, with order of elements reversed
`Param`	Algorithm parameter

Common symbols

$*$	Convolution (Equation 3.3)
$b(x,y)$	Local blobness
$B(x,y)$	Context enhanced blobness
G_σ	Gaussian kernel, at scale σ (Equation 3.5)
$G_\sigma^{(\mu,\nu)}$	(μ,ν)-th Gaussian derivative kernel at scale σ (Equation 3.6)
L	Image (Section 3.2)
m	Angular Fourier component (spherical harmonic) (Subsection 3.5.1)
$r(x,y)$	Local ridgeness
$R(x,y)$	Context enhanced ridgeness
(r,φ)	Polar coordinates
R_θ	Rotation matrix (Equation 3.4)
$\mathrm{R}_{\mathrm{ir},\theta}$	Diagonalized rotation matrix (Equation 3.18)
S	Similarity transformation matrix
(x,y)	Cartesian coordinates in two dimensions
θ	Rotation angle
σ_{ctx}	Scale of context for tensor voting, first step
σ_{ctx2}	Scale of context for tensor voting, second step
σ_{local}	Local filter scale
$\psi(x,y)$	Local orientation
$\Psi(x,y)$	Context enhanced orientation
Ω	Image domain (Section 3.2)

Chapter 3

$\langle \bullet \mid \bullet \rangle$	$\mathbb{L}^2$ inner product (Equation 3.2)
$\mathbf{b}$	Cartesian coordinate in two dimensions written as vector, $\mathbf{b} = (x,y)$
g_m	Gaussian m basis functions (Equation 3.41)

$h(x, y)$	Filter kernel
$h_\theta(x, y)$	Filter kernel, rotated by angle θ
$\mathbf{h}(x, y)$	Vector of basis filters of a steerable filter (Subsection 3.5.2)
$\mathbf{k}(\theta)$	Coefficients as function of rotation of a steerable filter (Subsection 3.5.2)
$l_{\varrho,\sigma_\mathrm{d}}$	Catheter profile function (Equation 3.13)
$L_\sigma^{(\mu,\nu)}$	(μ, ν)-th Gaussian derivative of image L at scale σ (Equation 3.7)
$r(\mathbf{b}, \theta)$	Local ridgeness in direction θ (Equation 3.16)
u_m	m-components of the image, i.e. $u_m = L * g_m$ (Subsection 3.5.6)
α_m	Weighting coefficients for g_m components constituting a filter kernel (Subsection 3.5.4)
σ_d	Device transfer function scale (Subsection 3.4.1)

Chapter 4

C_m	Complex encoded local ridgeness and local orientation (Equation 4.38)
$\mathbf{e}_1, \mathbf{e}_2$	Eigenvectors of a tensor (Equation 4.1)
$\mathrm{K}(\theta)$	Coefficient matrix for steerable voting field (Equation 4.29)
$\mathrm{L}(\theta)$	Coefficient matrix for steerable voting field expressed in m-components (Equation 4.33)
R'_θ	Rotation matrix for tensors (Equation 4.4)
$\mathrm{R}'_{\mathrm{ir},\theta}$	Diagonalized rotation matrix for tensors (Equation 4.5)
U	Output tensor field of tensor voting (Equation 4.22)
U_0,U_{-2},U_2	m-components of output tensor field of tensor voting (Subsection 4.3.2)
V	Steerable stick voting field (Equation 4.25)
V_θ	Steerable stick voting field, rotated over angle θ (Equation 4.25)
$\mathrm{V_B}$	Ball voting field (Equation 4.21)
$\mathrm{V_{cc}}$	Cocircularity tensor field (Equation 4.12)
$\mathrm{V_S}$	Medioni's stick voting field (Equation 4.18)
$W_m(x, y)$	Basis functions for steerable voting field (Subsection 4.3.2)
λ_1, λ_2	Eigenvalues of a tensor (Equation 4.1)

Chapter 5

$\mathcal{B}$	Set of electrode candidates (Subsection 5.3.1)
$c_{\mathrm{global}}(\mathcal{P})$	Global cost function on path $\mathcal{P}$ (Equation 5.38)
$\mathcal{C}$	Electrode group candidate (Subsection 5.3.3)
$\mathcal{E}$	Set of connections between electrode candidates (Subsection 5.3.2)
$\mathcal{G}$	Set of connections between paths (Subsection 5.2.6)
$\mathcal{P}$	Path (Subsection 5.2.1)
$\mathcal{P}_\mathrm{tip}$	Path representing a catheter tip (Subsection 5.3.4)
$\mathcal{Q}$	Set of paths (Subsection 5.2.1)
$s(\bullet)$	Certainty of image feature denoted by $\bullet$
$\mathcal{S}$	Super path (Subsection 5.4.1)
$\mathcal{W}$	Set of super paths (extension candidates) (Subsection 5.4.1)
$\theta_\mathrm{L}[i]$, $\theta_\mathrm{R}[i]$	Left-to-right resp right-to-left estimation of path direction at position i (Equation 5.11)
$\rho[i]$	Estimation of path curvature at position i (Equation 5.12)

Chapter 6

$\mathcal{B}$	Set of intensity values of all background pixels (Subsection 6.1.1)
CI	Confidence interval of 95% (Equation 6.5)
$\mathcal{F}$	Set of intensity values of all foreground pixels (Subsection 6.1.1)
HC	Histogram-consistency (Equation 6.4)
SBR	Signal-to-background ratio (Equation 6.1)

Contents

Chapter 1

Introduction

X-ray fluoroscopy imaging is frequently used for image guidance during minimally invasive medical procedures. During such interventions, guide wires or catheters are inserted into the patient and navigated through the body to the site of the problem. Some interventions can take a long time, especially electrophysiologic studies of the heart, which can last for several hours. Although X-ray is known to be harmful, no reasonable alternatives yet exist. It is therefore of interest to decrease the dose as much as possible. A lower dose, however, means deteriorating image quality. So image analysis techniques that are robust to noise are important. To automatically control the dose and to enhance image quality, the object of interest (e.g. the guide wire or catheter) has to be detected in the image.

The need for noise-robust image analysis techniques for detecting elongated structures, links to recent research in biologically inspired image analysis. The idea is to use so-called *context filters*, that model the continuation properties of elongated structures to communicate information over a larger spatial context. The expectation is that these techniques will lead to better performance regarding noise-robustness.

1.1 Aim of the project

The goal of this master project is the design of a method for automatic EP (electrophysiology) catheter detection in X-ray images that is robust to noise. The project restricts to the use of spatial context only, i.e. we only use information in a single frame. Other people have been working on tracking algorithms for elongated structures. Such algorithms need an initial position in an initial frame and then track the object over time (e.g. the work of Shirley Baert [4] for guide wire tracking).

The goal for the long term is to use the algorithms in this thesis in a contrast-to-noise ratio (CNR) based dose control system where the amount of dose is controlled by the visibility of the catheter. The findings of this work are also useful to enhance the visibility of the catheter in the image.

The master project was performed half-time at Philips Medical System (PMS, X-ray pre-development, Best), and half-time at the group Biomedical Image Analysis (BMIA, Eindhoven University of Technology, department of Biomedical Engineering). The reason for this construction is that BMIA has knowledge on new theoretical concepts for image analysis, which might lead to better methods for solving the difficult problem from practice at PMS.

1.2 Outline of this report

This thesis is built up as follows. In Chapter 2 the application of electrophysiology will be treated in more detail. Also, our approach to solve the problem will be described. In Chapter 3 methods to detect local oriented features in noisy fluoroscopy images will be introduced. In Chapter 4 methods will be introduced to enhance the local feature information using a context communication mechanism called *tensor voting*. In Chapter 5 the methods will be described to extract the EP catheters from the resulting feature images. In Chapter 6 the techniques will be evaluated. Finally in Chapter 7 we will draw the conclusions and pose recommendations.

Chapter 2

Application and Approach

This chapter introduces the different aspects of the problem. First, the application will be introduced in more detail. Then, the line of thought that forms the basis of the design decisions will be explained.

2.1 Cardiac electrophysiology procedures

The medical application we focus on is *cardiac electrophysiology* (EP). Cardiac electrophysiology procedures are performed to treat patients with *arrhythmias*, i.e. patients with abnormal heart rhythms.

Normally, electricity flows throughout the heart in regular, measured patterns. This electrical system is the basis for heart muscle contractions (Figure 2.1). Sometimes, however, the electrical flow is blocked or travels the same pathways over and over again, creating a kind of "short circuit" that disturbs the normal heart rhythm. There are two categories of arrhythmias: *bradycardia*, denoting a heart that beats too slow, and *tachycardia*, denoting a heart that beats too fast. The latter category is most common and can be divided in tachycardia originating in the heart's atria (*supraventricular tachycardia*), tachycardia originating in the AV node (junctional tachycardia), and tachycardia originating in the heart's ventricles (ventricular tachycardia). In some cases of tachycardia the problem is caused by a small rapidly firing spot (*ectopic focus*) that is triggering the abnormal rhythm. In other cases, the problem is caused by an undesired conduction path causing additional contractions of the heart.

Arrhythmias can not always be resolved by medication. Depending on the type of arrhythmias, a treatment by means of a minimally invasive *catheter ablation* procedure is an option. Catheter ablation is, apart from pacemaker implementation, the most common therapeutic cardiac electrophysiology procedure. It is most effective to treat supraventricular tachycardia and junctional tachycardia. Less frequently, the procedure is used to treat ventricular tachycardia.

The catheter ablation procedure involves insertion of one or more flexible thin tubes (*EP catheters*) through small skin incisions, usually in the groin, and threading them through blood vessels into the heart. The movement of the catheter through the body is guided using a real-time X-ray fluoroscopy imaging system. The EP catheters contain a number of electrodes used to make *intracardiac electrograms*, i.e. electrograms of the inside of the heart. Using these electrograms, the spot or conduction path causing the arrhythmias can be identified. A special EP catheter (*the ablation catheter*) emits radiofrequency energy to destroy the firing spot or to block the undesired conduction path. Often, the ablation procedure is sufficient to remedy the problem.

Catheter ablation is a complicated procedure, performed by specialists called *cardiac electrophysiologists*. Other terms for the same procedure are: *cardiac ablation*, *cardiac catheter ablation*, *radiofrequency ablation*, or simply *ablation*.

The next subsections offer more details on the EP catheters and the ablation procedure. More information can be found at [1] and more detailed information in [25].

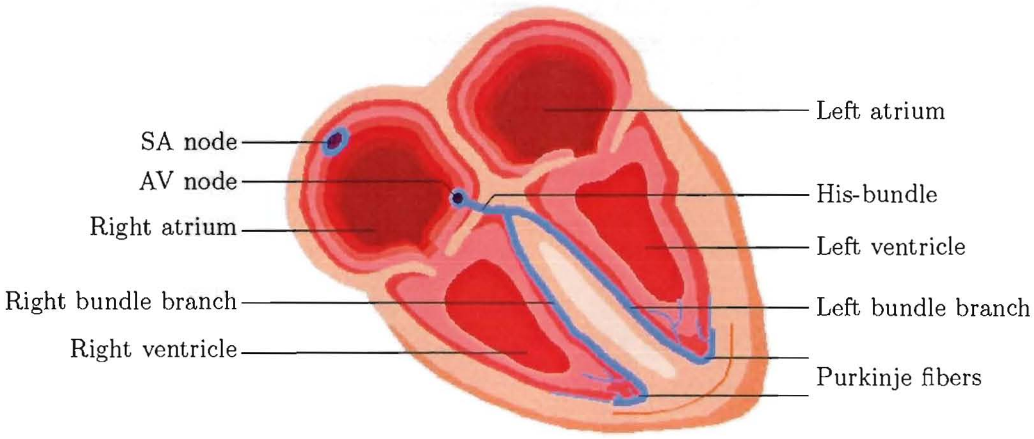


Figure 2.1: *The heart. The electrical signals that control the heartbeat are normally generated in the SA node. The electrical signal then spreads across both atria (upper chambers) of the heart causing contraction. The AV node receives the impulses from the SA node and controls the delayed contraction of the ventricles (lower chambers) via an electrical signal through the His-bundle and the Purkinje fibers. In case of arrhythmias, the electrical circuit is disturbed. For example, atrial fibrillation is an arrhythmias where the AV node is bombarded with chaotic and rapid impulses from the atria, causing the ventricles to beat rapidly and irregularly.*

2.1.1 EP catheters

There are many different types of EP catheters, all for specific situations. The two different main classes are EP catheters that are only used for mapping the inside of the heart (*mapping catheters*) and catheters that can also be used to do the ablation (*ablation catheters*).

Figure 2.2 shows photographs of two EP catheters. The material of the EP catheters is woven Dacron or a synthetic material. The length typically lies between 1.5 and 2.2 meters. On one side, the EP catheters have a *tip* with several electrodes. The number of electrodes varies for different EP catheters. They can contain 2 to 40 electrodes with 1 to 10 mm spacing. Usually the electrodes appear in pairs. Figure 2.3 shows some examples.

The ablation catheters have a special large platinum tip (4 to 10 mm) for the delivery of radiofrequency (RF) energy, to create a lesion inside the heart. The ablation tip also contains a temperature sensor, which is useful for controlling the size of a lesion during ablation.

All EP catheters have a handle on the other side used as a grip to steer the catheter through the body. At this end of the catheter, a connector is mounted to connect the catheter to special EP measurement equipment and to an RF generator in case of an ablation catheter. To ease navigation, the handle has a button that allows the electrophysiologist to remotely change the curvature of the EP catheter tip, see Figure 2.4.

2.1.2 The catheter ablation procedure

The ablation procedure involves the following steps:

1. Insertion of several EP catheters into the body through the femoralis vein (in the groin), the internal jugular, or subclavia vein (in the shoulder).

2. Propagation of catheters to the right heart atrium or ventricle guided by X-ray fluoroscopy images (Figure 2.5).

3. Pacing and cardiac mapping. The electrodes at the tip of the catheter are used for electrogram measurements at different locations of the myocardium inside the heart (this is called *cardiac mapping*). The resulting data pinpoints the location of the faulty electrical site. During mapping the electrophysiologist can use the EP catheter to *send* impulses to spots of the heart (this is called *pacing*), in order to instigate tachycardia to get useful measurements.

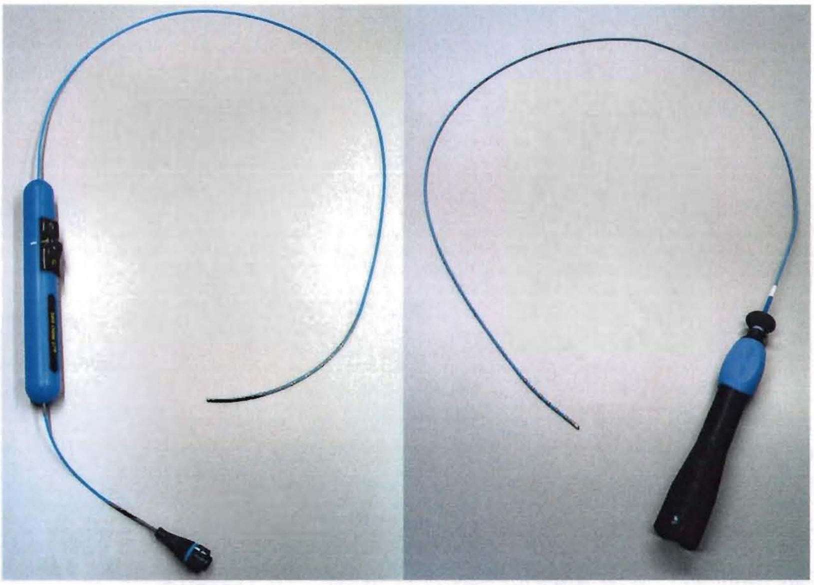


Figure 2.2: *Two examples of EP catheters. The left one is an EP catheter with 20 electrodes placed in pairs. The side with the electrodes is inserted in the body. The large handle is the grip to hold the EP catheter. It contains a button to steer the tip (Figure 2.4) to navigate the catheter through the vascular system and the heart. The large black end at the bottom is a connector for connecting the EP catheter to EP measurement equipment. The EP catheter at the right contains 6 electrode and an ablation tip. The connector of this catheter is located at the bottom side of the handle.*

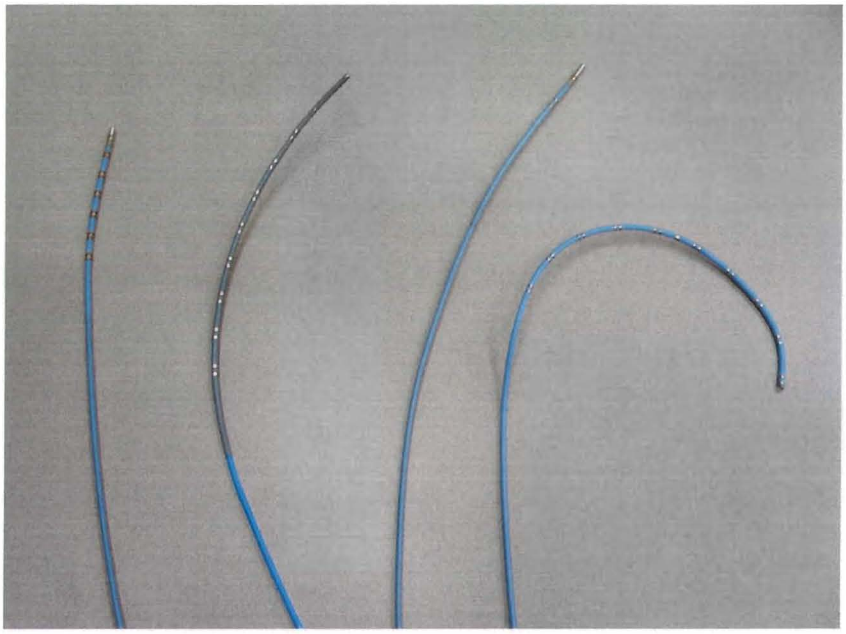


Figure 2.3: *A few examples of EP catheter tips with different electrode configurations. The first and third catheters (counted from the* ***left****) are ablation catheters, recognizable by their large tip.*

4. Treatment planning: at this stage, the electrophysiologist decides how to treat the arrhythmias. He can not do this before the procedure, because not enough information is available at that stage.

5. Pacing and cardiac mapping. Aim of this step is to find the spot that should be ablated and

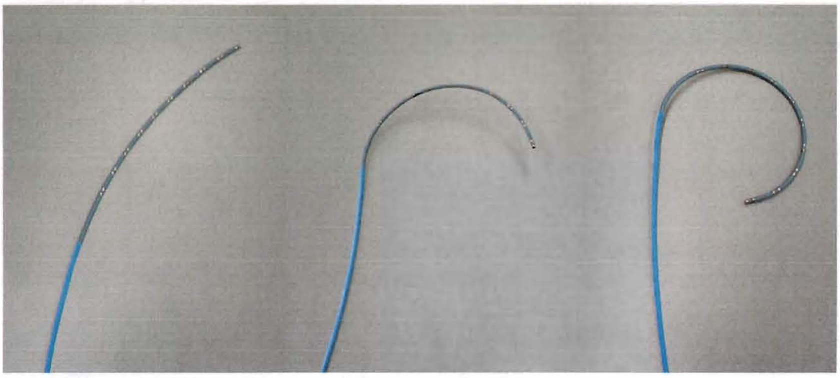


Figure 2.4: *Steerability of the EP catheter tip. Using a button on the end of the catheter that is outside the body, the curvature of the tip of the catheter can be changed.*

to check correctness again.

6. Catheter ablation. The ablation tip is positioned at the problematic location of the myocardium. Radiofrequency energy is sent to the tip of an ablation catheter to create a lesion at a spot (*focal ablation*), or to create a lesion along a line (*linear ablation*). The lesion results in scar tissue that will no longer conduct electrical signals.
7. Check. Again intracardiac electrograms are measured and pacing is performed to check whether the arrhythmias has disappeared. If not, the ablation is not complete and some steps need to be repeated.
8. Finally, the interventional devices are removed and the small incision is closed.

A catheter ablation procedure is time-consuming. It *typically takes 2 to 4 hours*. During the whole procedure X-ray fluoroscopy imaging is necessary resulting in a high cumulative radiation dose. The high accumulating dose is especially problematic for the medical staff that is exposed to radiation the whole day. Especially the mapping of the electrical activation inside the heart is time-consuming. Currently, mapping is done manually, which means that a map of the heart is to be formed in the "human memory" of the electrophysiologist.

For these reasons, important research topics related to catheter ablation procedures are:

- *Automatic cardiac mapping.* The use of computer memory to form a map instead of human memory will lead to a considerable time gain. For this purpose, besides the recording of electrograms, the computer needs information about the 3D position of the electrodes of the catheters to reconstruct a 3D map of the heart. The 3D positions can be obtained using an active tracking method (i.e., the catheter contains a sensor that measures the location of the catheter in a magnetic field) or image processing using biplane fluoroscopic images [27] (i.e., in X-ray images acquired at two different angles, the positions of the catheters can be determined via stereoscopic calculations).
- *Dose reduction.* The dose can be reduced by optimizing the X-ray imaging system. Last but not least, image analysis can be useful to accomplish further reduction of dose.

The focus of this project is image analysis for dose reduction. Below, X-ray dose reduction will be treated in more detail.

2.2 X-ray dose reduction for cardiac electrophysiology

During cardiac electrophysiology procedures, it is especially important for the electrophysiologist to clearly see the EP catheters in the X-ray image *and* the border of the heart as a reference. So, the aim is to reduce the dose in such a way that these features are still sufficiently visible.

The temporal response also need to be fast enough in order to navigate through the body with the catheters. In the examples of Figure 2.5, the EP catheters are clearly visible, because these images are acquired using a relatively high dose. The border of the heart is more difficult to see, especially in a single frame. The border is better visible when watching the image sequence due to heart motion.

The importance of dose reduction for electrophysiology is stated in [8], which describes a joint project between The University of Leeds and Philips Medical Systems. The following adaptations to the X-ray imaging system are proposed to reduce the dose:

- *Pulsed* fluoroscopy is used to get a better temporal response and to reduce the total dose. The frame rate has to be at least 12.5 fps, otherwise the positioning of catheters becomes hard, because of the time delay.
- An additional spectral beam filtration is utilized to reduce noise levels.
- A hierarchical dose regime is proposed, with three settings: "ultra low", "low", and "normal". The ultra low setting is sufficient most of the time. For demanding sub-procedures the operator can switch to the low or normal setting.

This results in a huge dose reduction (typically 80% reduction for an ablation procedure).

2.2.1 The use of image processing

To further reduce the dose, advanced image processing techniques can be useful. Image processing can be employed in different ways. First, it can be used to enhance the visibility of structures of interest. At a low dose, it can be tiring for the operator to still see, for instance, the EP catheter. If we can detect the position of the catheter in the image, this information can be used to highlight the EP catheter on the monitor that is used for navigation.

Second, the required dose level depends on patient thickness and on the orientation of the X-ray detector. Therefore, an automatic dose control mechanism is desirable to keep the dose on a level that is just sufficient. The "old way" of controlling dose is by measuring average contrast in the X-ray image, and keeping this at a fixed level. New ideas for dose control within Philips are based on controlling the dose, such that the structure of interest has a certain visibility [34]. This visibility can be expressed by the *contrast-to-noise ratio* (CNR), defined as the ratio between the contrast of the structure of interest relative to its neighborhood and the estimated noise level in the neighborhood of the image. For this purpose, image processing techniques are needed to detect the structure of interest. In this specific application, the structure of interest is the EP catheter[1].

Figure 2.6 gives a schematic overview of the fluoroscopy imaging system, including the image processing blocks.

2.2.2 Project focus: detection of EP catheters

The focus of this master project is the detection of the EP catheters, i.e. the *image analysis* block in Figure 2.6. Because the required user intervention should be as low as possible, we want to design an EP catheter detection algorithm that is *fully automatic* and not based on one or several manually pointed seeds. The final goal is to lower X-ray dose. It is therefore important that the detection algorithm is robust to noise.

For clinical practice, detection of the *tip* of a catheter is probably most important, since this is the part of the catheter that is essential for navigation through the body. However, at the time the project started it was not clear what part of the catheter is important. Thus, we designed an algorithm to detect the entire catheter. The detection of the entire catheter is more error-prone than the detection of the tip only. In the evaluation we will therefore investigate both the feasibility of extracting only the tip, and the feasibility of extracting the entire catheter.

---

[1] Note that also the border of the heart is important. Controlling the dose using the EP catheter visibility could lead to an invisible heart border. This could become a problem in future.

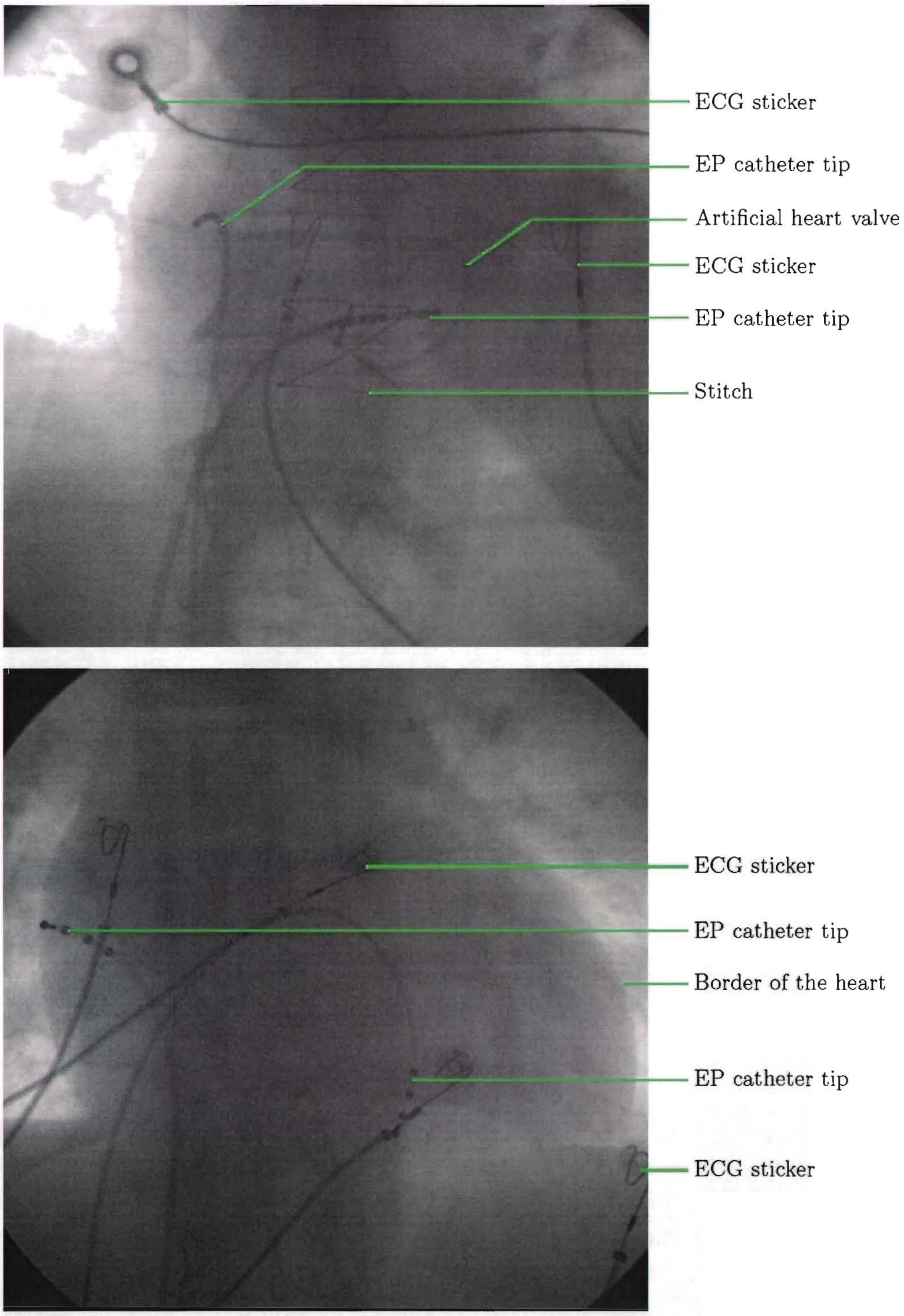


Figure 2.5: *Two examples of typical EP X-ray images acquired during real clinical interventions.*

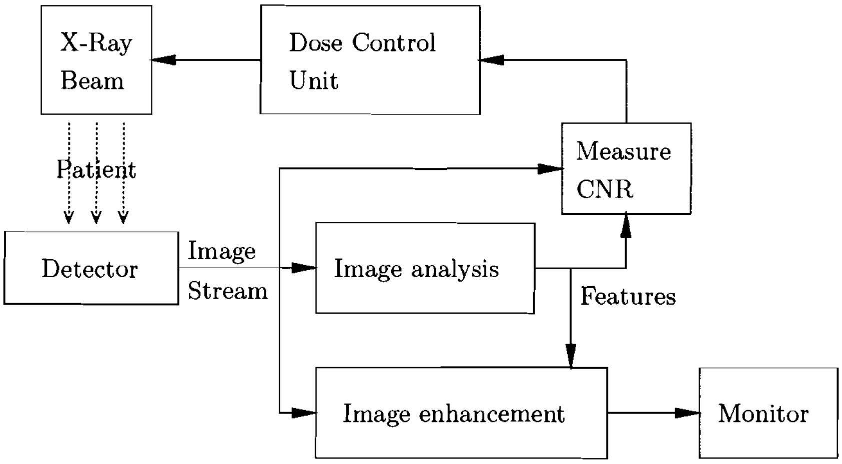


Figure 2.6: *Schematic overview of the fluoroscopy imaging system. The patient is X-rayed, the resulting image stream is analyzed to detect features of interest (in our case the EP catheters). The detected features are used for measuring the visibility in the image to automatically adjust the dose to the desired level. The detected features can also be used to enhance image quality and to improve the catheter visibility.*

It is important to realize that X-ray fluoroscopy requires real-time imaging, so a real-time algorithm is needed in practice. Although this should be taken into account, the speed of the algorithm will not be our main concern, but rather its effectiveness. If the algorithm proofs to be useful, it can be made more efficient later on. Some considerations on speed will be included in the chapter on evaluation.

We will focus on the use of spatial information, i.e. information contained in a single frame. Besides spatial information, temporal information appears very useful as well, but this lies outside the scope of this project.

2.2.3 Characteristics of EP images

For proper detection of objects in EP images, knowledge on the characteristics of the images and the EP catheters is important. In this subsection, the qualitative knowledge on the images and EP catheters will be summarized. These characteristics are important to mention explicitly, because they form the basis of a lot of algorithm design decisions. The most important characteristics of the images are:

- Low Signal-to-Noise ratio. The dominant type of noise in fluoroscopy is quantum noise, which exhibits a Poisson distribution. For Poisson noise the variance is proportional to the gray value intensity. However, because of image pre-processing steps applied to the data by the X-ray imaging system[2], the variance increases at lower intensity values and decreases at higher intensity values. The noise spectrum does not correspond to the spectrum of white noise due to the modulation transfer function and the temporal resolution of the system. More information on X-ray specific noise characteristics can be found in [26].
- Often, more catheters are visible simultaneously. There are also a lot of other elongated objects in the image, for instance stitches of an earlier intervention and ECG stickers. These objects make the detection more difficult, especially due to crossings.
- It is important to realize that the 2D X-ray images we work with, are a projection from 3D to 2D. Consequently, catheters can have sharp bends in a 2D image. Also, some characteristics of the EP catheter can be hidden when the catheter points in the direction of the X-ray beam or towards the detector.

---

[2] Various image pre-processing steps are performed, but the most important effect is caused by *white compression*: this is a grayscale transformation that stretches the dark gray values and compresses the bright values.

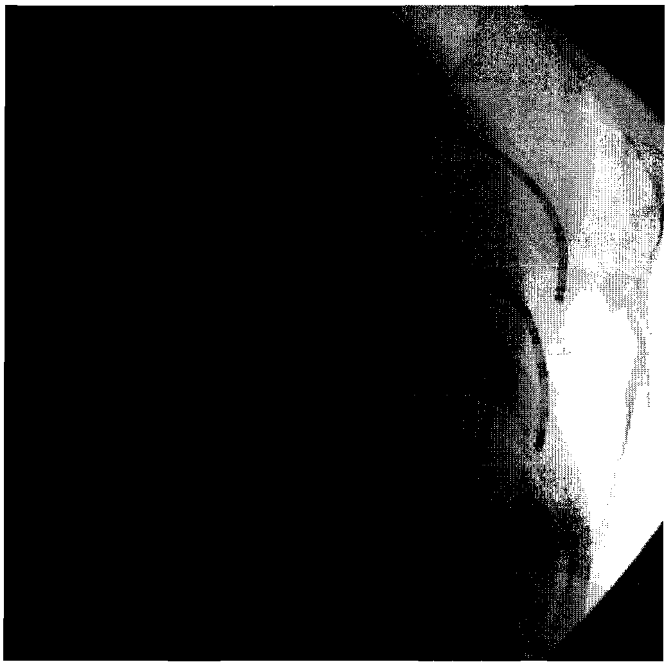


Figure 2.7: *Some different EP catheters as they appear in an X-ray image.*

EP catheters in X-ray images (Figure 2.7) exhibit a lot of properties that can be used for proper detection. We know that:

- Catheters (and guide-wires) appear in the fluoroscopy image as dark lines with almost constant thickness.
- The catheter will have a limited amount of total curvature and length (at least in 3D, but in 2D this is usually the case, too).
- On one side of the image the catheter will disappear out of the field of view.

EP catheters always contain electrodes. On the electrodes we can utilize the following features:

- The electrodes on a catheter appear as dark blobs that are approximately isotropic, except for the tip of an ablation catheter, which is more elongated.
- The electrodes are always situated on the line of the catheter.
- The electrodes on the EP catheter have a specific spacing. The exact spacing depends on the type of EP catheter.

Note that the electrodes are in some situations not visible, due to the projection from 3D to 2D.

The points above only characterize still images, because the project is restricted to still images. However, in a later stage the use of temporal information is recommendable. In EP image sequences, there are two causes of motion, namely the motion caused by the patient (beating of the heart, breathing), and the motion caused by the electrophysiologist positioning the EP catheters. Generally, objects have a limited amount of movement between subsequent frames. The patient motion also has a certain periodicity. This information can be used for tracking EP catheters over time.

2.3 Approach

The problem of detecting EP catheters *without* an initial seed, implies that we have to search in a fairly large search space. Iterative search algorithms, like *snakes* [24], are therefore not suitable.

These methods are only appropriate if we already have an estimation of the catheter position[3].

A more appropriate strategy for this "global search problem" is to apply global operations, in such a way that the structures of interest *emerge* from the data. In other words, we want to amplify structures of interest and attenuate other structures. Then, we can extract the object of interest by extracting the most salient features that emerged from the data.

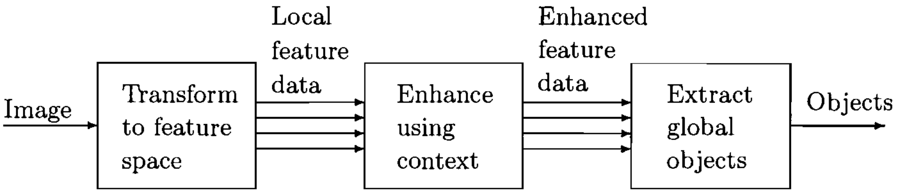


Figure 2.8: *Framework of our method. See text for details.*

Figure 2.8 shows the general framework of our EP catheter extraction process. The method is divided in three main stages.

1. In the first stage, *transform to feature space*, we detect local image features using filter operations that encompass knowledge on the *local image structure* of the features of interest. The result is a number of *local feature images*, in which the pixel values represent the *likelihood* that a certain feature is present at that position of the image. Different features are possible depending on the application. The local image features we are especially interested in are:

   - *Ridges* for detecting the EP catheters, which locally exhibit a ridge shape in the intensity landscape of the image.
   - *Blobs* for detecting the electrodes on the EP catheters, which locally exhibit a concave shape in the intensity landscape of the image.

   We call the resulting feature images the *local ridgeness* and *local blobness* images. Note that in this step, the pixels are not really classified as being part of a ridge or a blob. Only a likelihood measure is calculated. The reason is that local image features are considered too unreliable, especially in noisy images. This part is described in Chapter 3.

2. In the previous stage, only local image features are taken into account. The aim of the next stage, *enhance using context*, is to improve the local feature data by taking a larger extent of the object of interest into account. A *context communication scheme* enables local feature data to communicate over a larger spatial neighborhood, using prior knowledge on the object of interest. The output is a number of *context enhanced* feature images.

   The principle is as follows. If a position (x, y) in the image has a large local ridgeness value, and if also a lot of pixels in the neighborhood (the context) exhibit large ridgeness values, then it is more likely that (x, y) belongs to a line structure (e.g. an EP catheter). If, on the contrary, no other pixels in the neighborhood have a large ridgeness value, then it is less likely that (x, y) belongs to a line structure. For elongated structures like EP catheters, useful prior knowledge is that catheters generally exhibit low curvature.

   Our method for context enhancement is described in Chapter 4.

3. In the last stage, *extract global features*, the context enhanced feature images generated by the previous step are used to extract the *objects of interest* in the image. The image often contains other objects with similar characteristics, which are also present in the context enhanced feature images. Therefore, the aim of this step is to finally decide where the EP catheter is located. This is accomplished by searching for object *candidates* in the image, and then using application specific knowledge to select the best candidate. We will frequently refer to this step as the *high-level extraction* step. This part is described in Chapter 5.

---

[3] Note that this type of methods might be very appropriate for the temporal tracking of the catheter

Chapter 3

Local Feature Detection

This chapter describes the first step in our EP catheter detection procedure. We will design methods to detect the local image features of interest: ridges, orientation of ridges, and blobs (Figure 3.1).

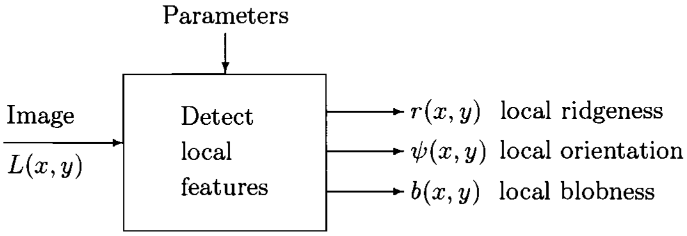


Figure 3.1: *Black-box of local feature detection. The X-ray image is the input of the box. As output feature images are generated for orientation, ridgeness, and blobness.*

3.1 Overview

Figure 3.2 shows an overview of the local feature detection method. The purpose of the first step, *background equalization*, is to remove background structures that can disturb the detection of the features of interest. Disturbing background structure are plateau edges and large scale variations in intensity.

The structures we are interested in are contained in the *differential structure* of the image. Therefore, we introduce in the next step the Gaussian derivatives of the image. Gaussian derivatives are chosen because they offer a lot of interesting properties: the Gaussian kernels are x, y-separable, they are steerable, they have a clear mathematical meaning, and they only have one free parameter (scale), which has a clear interpretation [37, 12].

The Gaussian derivatives are used to calculate the features we are interested in: local ridgeness, orientation of the local ridge, and blobness. The calculation of these features is performed pixel-wise, i.e. for every pixel position the measures are calculated solely using the derivatives at that pixel position.

The following sections will treat the different parts in more detail. Special attention will be paid to ridge detection using steerable filters. We will discuss the possibilities to make the filter more elongated to get better response on elongated line structures.

3.2 Definitions

The following definitions will be used throughout this chapter.

- Two-dimensional image: $L(x,y)$ with $(x,y) \in \Omega$ the domain of the image, $\Omega \subset \mathbb{R}^2$.

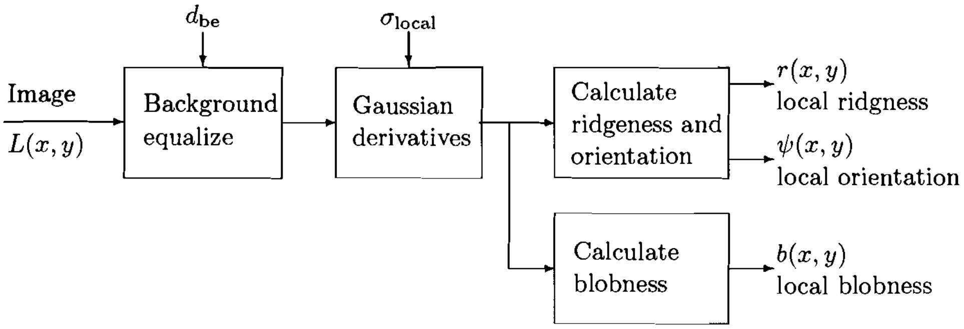


Figure 3.2: *Schematic overview of the local feature detection part.*

- $\mathbb{L}^2$ function space. All functions $f : \Omega \mapsto \mathbb{C}$ that are square integrable, i.e.

$$\int_\Omega f(x,y)\,\overline{f}(x,y)\,dx\,dy, \tag{3.1}$$

exists and is finite. In this chapter, we assume our images L and our filters to be in $\mathbb{L}^2$ function space.

- $\mathbb{L}^2$ inner product of two $\mathbb{L}^2$ functions a and b

$$\langle a\,|\,b\rangle = \int_\Omega a(x,y)\,\overline{b}(x,y)\,dx\,dy. \tag{3.2}$$

- Convolution

$$(f * g)(x,y) = \int_{\mathbb{R}^2} f(x',y')\; g(x-x',y-y')\;dx'\,dy'. \tag{3.3}$$

- Rotation matrix

$$\mathrm{R}_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}. \tag{3.4}$$

- Gaussian kernel with scale σ

$$G_\sigma(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}. \tag{3.5}$$

- (μ,ν)-th Gaussian derivative kernel with scale σ

$$G_\sigma^{(\mu,\nu)} = \left(\frac{-1}{\sqrt{2}\,\sigma}\right)^{\mu+\nu} H_\mu\Big(\frac{x}{\sqrt{2}\,\sigma}\Big)\, H_\nu\Big(\frac{y}{\sqrt{2}\,\sigma}\Big)\, \frac{1}{2\pi\sigma^2}\, e^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{3.6}$$

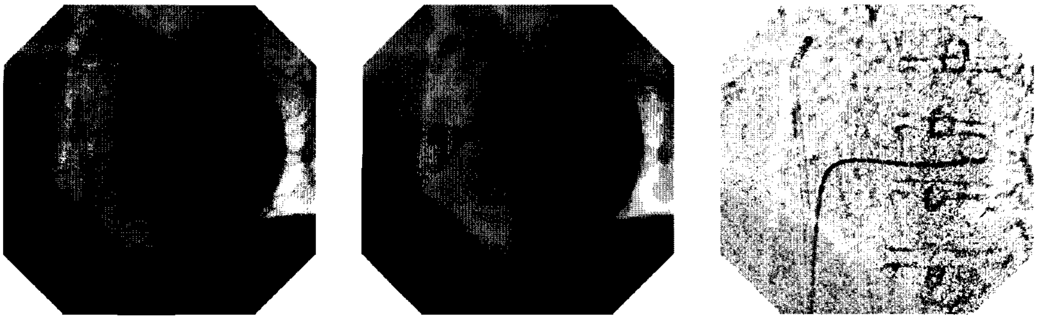
with μ the order of derivative in x-direction, ν the order in y direction, and $H_\mu(x)$ the ν-th Hermite polynomial.

- (μ,ν)-th Gaussian derivative of image L at scale σ

$$L_\sigma^{(\mu,\nu)}(x,y) = (L * G_\sigma^{(\mu,\nu)})(x,y). \tag{3.7}$$

3.3 Background equalization

Some features in the background of the EP catheters images are quite disturbing when detecting blobs and ridges. Therefore, a preprocessing step is performed to remove undesirable structures in the background. Especially *plateau edges*, edges forming a boundary between parts of the image with a different background gray value, can be disturbing, because a ridge detector gives a high response also on these edges.



(a) (b) (c)

Figure 3.3: *Background equalization. (a) Original image L. (b) Background image L_{backgr}, obtained by a morphological closing with a disc-shaped structure element. (c) Resulting background equalized image L_e. Note that the plateau edges almost disappear, while the EP catheter is still clearly visible.*

Our strategy is similar to the one in [27]. A morphological *closing* operation with a disc-shaped structure element is applied on a slightly blurred version of the input image L (with $L(x,y) \geq 0 \; \forall (x,y) \in \Omega$), to obtain an image that only contains background structure

$$L_{\text{backgr}} = \text{close}(L * G_{\sigma_b}, S_{d_{\text{be}}}), \tag{3.8}$$

where S is a disc-shaped structure element with pixel diameter d_{be}, which should be about two times the pixel width of an EP catheter, and G_{σ_b} is the Gaussian kernel with scale σ_b so that $L * G_{\sigma_b}$ is a slightly blurred version of image L. The reason to apply this blurring is to obtain a somewhat smoother background structure image. A closing operation is a dilation followed by an erosion. The dilation will remove the smaller structures, including the EP catheters. The plateau edges will be shifted due to this morphological operation. But after the erosion with the same structure element, the plateau edges will be shifted back to their original position.

L_{backgr} only contains the background structure that we want to remove. The background equalized image is obtained by dividing the original image by L_{backgr}, i.e.

$$L_{\text{e}} = \frac{L}{L_{\text{backgr}}} \max_{\Omega}(L), \tag{3.9}$$

where the term $\max_{\Omega}(L)$ is added to get L_e within the same range as L.

We use division because X-ray images exhibit shadows that are due to the exponential decay of luminosity as function of object opacity. Suppose the X-ray beam has an intensity I_0. The beam has to penetrate the catheter and other objects in the background. The X-ray intensity I at the detector is given by

$$I = I_0 \cdot \beta_{\text{c}} \cdot \beta_{\text{b}} \quad \text{with } 0 \leq \beta_{\text{c}} \leq 1 \text{ and } 0 \leq \beta_{\text{b}} \leq 1, \tag{3.10}$$

where β_{c} is the attenuation coefficient of the catheter, and β_{b} is the attenuation coefficient of the background, see Figure 3.4. From this equation we can see that division by β_b cancels out the effect of the background structures. This is exactly what we do in Equation 3.9 since $L_{\text{backgr}} \approx I_0\beta_{\text{b}}$ and $I_0 \approx \max_{\Omega}(L)$.

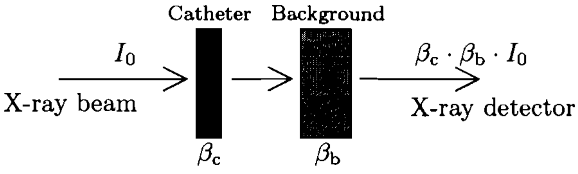


Figure 3.4: *Model of X-ray imaging system with catheter and background.*

It should be noted that we neglect the postprocessing that is performed by the X-ray imaging system, for instance the white compression. For optimal performance of the background equalization we should compensate for this. However, the post-processing that is performed depends on the settings and type of the X-ray imaging system and therefore we do currently not know exactly how to compensate for it. It is advisable to take this into account in future.

Figure 3.3 shows an example of background equalization. The plateau edges disappeared, and the catheter is still clearly visible. Henceforth, we will only use the background equalized version of the image. For the ease of notation we will refer to this image as L instead of L_{e}.

3.4 Ridge detection

In this section we treat the concepts that are necessary to understand ridge detection. First we create a model of the catheter, which will be used to design optimal ridge detection filters for catheters. Then, we describe the most straightforward approach to detect ridges.

3.4.1 Catheter profile function

In an X-ray image we observe the shadow of the catheter. Assuming that the catheter is perfectly round and made of homogeneous material, the one-dimensional profile function of the shadow of a catheter is given by

$$l_{\varrho}(x) = \begin{cases} 1 - \sqrt{1 - \frac{x^2}{r^2}} & \text{if } |x| \leq r; \\ 1 & \text{if } |x| > r, \end{cases} \tag{3.11}$$

where the catheter has a radius ϱ, and the gray values are normalized between 0 and 1. This is the “raw” profile function of an ideal catheter. The X-ray device has a certain transfer function, which can be approximated by a Gaussian kernel with scale σ_{d}. So the observed catheter profile function according to this model is

$$l_{\varrho,\sigma_{\mathrm{d}}}(x) = (l_{\varrho} * G_{\sigma_{\mathrm{d}}})(x). \tag{3.12}$$

Figures 3.5a-b show an example from a real X-ray image, and Figures 3.5c-d show the one-dimensional profile functions l_r and $l_{\varrho,\sigma_{\mathrm{d}}}$.

Assuming that the catheter is straight (i.e. has zero curvature), the two-dimensional catheter profile function is given by (see Figure 3.5e)

$$l_{\varrho,\sigma_{\mathrm{d}}}(x, y) = (l_{\varrho} * G_{\sigma_{\mathrm{d}}})(x). \tag{3.13}$$

A catheter can have any orientation[1]. The catheter profile function $l_{\varrho,\sigma_{\mathrm{d}}}$ can be rotated over an angle θ as follows

$$l_{\varrho,\sigma_{\mathrm{d}}}(\mathbf{b}; \theta) = l(\mathrm{R}_{\theta}^{\mathrm{T}}\mathbf{b}), \tag{3.14}$$

where $\mathbf{b} = \left(\begin{smallmatrix} x \\ y \end{smallmatrix}\right)$ and $\mathrm{R}_{\theta} = \left(\begin{smallmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{smallmatrix}\right)$ is the rotation matrix.

3.4.2 Ridge detection strategy

The general strategy to detect features in images is by *template matching.* A template is created that models the structure to be detected. The cross-correlation of the image with the template is calculated. The position with the highest cross-correlation has the best match with the structure modelled by the template.

The same strategy is applicable for ridge detection. We need to design a filter kernel $h(x, y)$ that is in fact a template of a catheter. The image L is convolved with the filter kernel, $L * h$. The resulting values are an indication for the presence of a catheter.

For appropriate detection of ridges, it is important to fulfill the following invariants:

---

[1] Note the difference between *orientation* and *direction*. An orientation has a 180° symmetry, i.e. if we rotate by 180° we get the same orientation, while a direction has a 360° symmetry. For instance, a point on a straight line is a 180° symmetric feature, while the end-point of a line is a 360° symmetric feature.

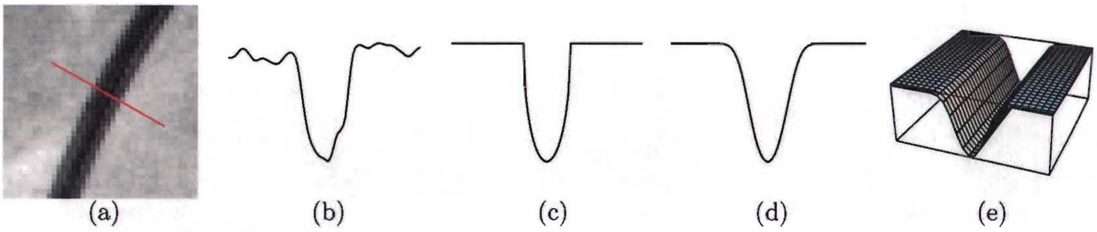


Figure 3.5: *Local one-dimensional model of a catheter shadow. (a) Example of a X-ray image of a catheter with low noise. (b) Intensity landscape profile orthogonal to the catheter. (c) Ideal profile catheter profile function* l_r. *(d) Gaussian blurred version of ideal profile* l_{ϱ,σ_d}, *modelling the device transfer function. (e) two-dimensional luminosity function for a straight catheter shadow.*

- Invariance under *translation*: the position of a ridge in an image should not influence the result. This property already holds since we use linear convolution.
- Invariance under *rotation*: a rotation of the entire image should not influence the result. This implies that in principle the ridge template needs to be matched with the image in *all orientations.*
- Invariance under *grayscale translation*: adding a "DC-component" to the image (i.e. $L'(x, y) = L(x, y) + a$, with a an arbitrary real value) should not influence the result. This results in the requirement that the filter kernel should fulfill $\int_{\mathbb{R}^2} h(x, y)dx\,dy = 0$. In this way we only look at the *differential structure* of the image [37, Chapter 6].

Ideally we would like to fulfill the following invariants, too:

- Invariance under *noise*: adding noise should not influence the result. In practice, we can only try to be as robust to noise as possible.
- Invariance under *scaling*: changing the settings of the X-ray detector (for instance by moving the detector, by changing the II mode, or by changing image resolution) should not influence the result. This implies the need for a multi-scale approach. However, we will not adopt such an approach for reasons that will be discussed in Section 3.7.

In Figure 3.5d one can see that the catheter profile function perpendicular to the ridge is locally similar to a parabola, i.e. $l_{\text{local}}(x) \simeq x^2$. Therefore, the use of a second order Gaussian derivative, along the axis orthogonal to the ridge is the most natural choice. Tangent to the ridge, the filter should take a certain neighborhood into account. The weight in the center of the filter kernel should be largest, and it should gradually decay. Therefore, a natural choice is to use a Gaussian kernel along this axis. This leads to the following filter kernel as an example of an appropriate ridge detector

$$h(x, y) = G^{(2)}_{\sigma_x}(x)\, G_{\sigma_y}(y), \tag{3.15}$$

where $G^{(2)}_{\sigma_x}$ denotes the second order Gaussian derivative with scale σ_x. This filter is called the *anisotropic second order Gaussian derivative.*

The filter has two free parameters. The scale parameter σ_x should be optimized with regard to the width of the line structures to be detected. The second parameter, σ_y, specifies the *elongatedness* of the filter (Figure 3.6). For the selection of this scale there are two counteracting effects to keep in mind. Making the filter more elongated will lead to a stronger response and a more precise orientation on a straight line structure. On the contrary, it will lead to a worse response if the line structure is curved, because then the filter does not "fit" onto the structure. A compromise between these two extremes needs to be found.

Detection of ridges with an orientation θ in an image L can now be achieved by

$$r(\mathrm{b}, \theta) = (L * h_\theta)(\mathrm{b}) \quad \text{with } h_\theta(\mathrm{b}) = h(\mathrm{R}_\theta^{\mathrm{T}}\mathrm{b}). \tag{3.16}$$

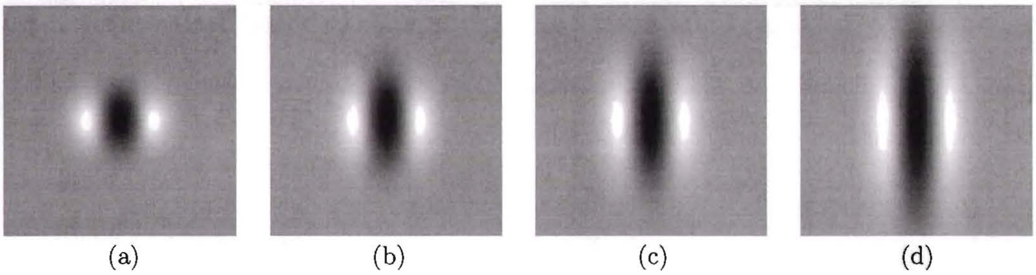


Figure 3.6: *Anisotropic Gaussian derivatives. (a) $\sigma_y/\sigma_x = 1$. (b) $\sigma_y/\sigma_x = 1.5$. (c) $\sigma_y/\sigma_x = 2$. (d) $\sigma_y/\sigma_x = 3$.*

To detect ridges in all orientations, one should calculate $r(\mathbf{b}, \theta)$ for a *sufficient* number of orientations $\theta \in [0, \pi)$. In theory infinitely many orientations are needed (to ensure rotation invariance), in practice the number of orientations depends on the application. The resulting ridgeness responses can be handled in two ways:

- As a stack of images, where each image in the stack represents the ridgeness response for a certain orientation. So the orientation is considered as an additional dimension. This concept is called an *orientation score*. The practical drawback of this representation is the huge amount of data to store and to process. The advantage is that more information is available, for instance at crossings of two line structures. Also, for a special class of kernels reconstruction of the original image from the orientation score is possible [22, 9].
- Alternatively, we can take the maximum ridgeness response for every pixel position, and keep the corresponding orientation, i.e.

$$\begin{aligned} r(\mathbf{b}) &= \max_{\theta} r(\mathbf{b}, \theta), \text{ and} \\ \psi(\mathbf{b}) &= \arg\max_{\theta} r(\mathbf{b}, \theta), \text{ for all } \mathbf{b} = \left(\begin{smallmatrix} x \\ y \end{smallmatrix}\right) \in \Omega. \end{aligned} \tag{3.17}$$

  In this way, for every pixel position the response is kept in the orientation that is most likely to be a ridge. We use this approach for the EP catheter detection problem.

Calculation of the ridgeness in many orientations is computationally unattractive, because a lot of convolutions with the image need to be calculated. Fortunately, there exists a method to do this in a more efficient way: the use of *steerable filters*. This is the subject of the next section.

3.5 Steerable filters for ridge detection

As stated before, calculating filter responses in many different orientations is time-consuming. For some filters, however, it is possible to use a more efficient method to calculate filter response in any desired orientation θ, without the need to explicitly convolve the image with the ridge detection kernel rotated by angle θ. Such filters are called *steerable filters*. The method has been introduced to image analysis by Freeman et al. [14], but the mathematical concepts originate from group theory [16].

3.5.1 Basic principles from group theory

To understand which filter kernels are steerable, we introduce some terminology from group theory[2]. A group G is a finite or infinite set of elements with a binary group operation that fulfills the properties of closure ($ab \in G$ for all $a, b \in G$), associativity ($(ab)c = a(bc)$ with $a, b, c \in G$), identity (there is a $e \in G$ such that $ae = ea = a$ for all $a \in G$), and inverse (for all $a \in G$ there

---

[2] Note that in this work we do not at all aim at giving a complete and thorough survey of group theory. To understand this work, a full understanding of group theory is not necessary. We will only try give the reader an intuition of the underlying principles that originate from group theory. For an introduction into the topic see [16])

exists a $a^{-1} \in G$ such that $a^{-1}a = e$). Abelian groups also fulfill the property of commutativity ($ab = ba$ for all $a, b \in G$).

In group theory, the 2D rotation matrices constitute the SO(2) group. SO(2) stands for *Special Orthogonal* 2D matrices. "Orthogonal" indicates that $\mathrm{R}^{\mathrm{T}} = \mathrm{R}^{-1}$ holds. "Special" indicates that $\det \mathrm{R} = 1$, i.e. we have *proper* rotations (no mirroring).

In order to detect features in any orientation, we have to rotate the filter kernels. For this purpose, we define a function space that contains our filter kernels and images as elements, on which the rotation group can act. We require all images and filter kernels to be member of the $\mathbb{L}^2$ function space (see Section 3.2), implying that the $\mathbb{L}^2$-norm $\langle f \,|\, f \rangle$ (Equation 3.2) exists and is finite.

To rotate $\mathbb{L}^2$ functions one has to find a group homomorphism, i.e. a map preserving the group product, between the rotation / SO(2) group and the group of unitary transformations in $\mathbb{L}^2$. The function space $\mathbb{L}^2(\Omega, dx\,dy)$ fulfills the criteria of an infinite-dimensional separable Hilbert space [2]. Any element of a Hilbert space can be expressed by a basis. So the unitary transformations can be expressed as unitary matrices (i.e. matrices that fulfill $(\mathrm{M}^{\mathrm{T}})^* = \mathrm{M}^{-1}$) with respect to such an enumerable basis.

The rotation matrix R_θ (Equation 3.4) is an example of such a unitary matrix. The matrix R_θ is a *reducible representation* of the group. Reducible means that the space, in which this representation acts falls into invariant subspaces under group transformations. By choosing an appropriate basis, we can find a representation made up by irreducible representations, i.e. we can find a basis, in which the irreducibility becomes apparent. For the group constituted by R_θ this is

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \underbrace{\begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}}_{\mathrm{R}_\theta} = \underbrace{\frac{1}{\sqrt{2}}\begin{pmatrix} -i & i \\ 1 & 1 \end{pmatrix}}_{\mathrm{S}} \underbrace{\begin{pmatrix} e^{-i\theta} & 0 \\ 0 & e^{i\theta} \end{pmatrix}}_{\mathrm{R}_{\mathrm{ir},\theta}} \underbrace{\frac{1}{\sqrt{2}}\begin{pmatrix} i & 1 \\ -i & 1 \end{pmatrix}}_{\mathrm{S}^{-1}} \begin{pmatrix} x \\ y \end{pmatrix}, \tag{3.18}$$

where S is a unitary similarity transformation matrix. The columns in S are the vectors spanning the basis, in which the irreducibility of the group becomes apparent in the form of $\mathrm{R}_{\mathrm{ir},\theta}$, i.e. these basis vectors form invariant subspaces under rotation.

Since the SO(2) group is an Abelian group, any irreducible representation of the group is one-dimensional. For SO(2) they are always of the form

$$\mathrm{R}_{m,\theta} = \left(e^{im\theta}\right) \quad \text{with } m \in \mathbb{Z}, \tag{3.19}$$

where $\mathrm{R}_{m,\theta}$ is a 1×1 unitary matrix that is homomorphic to SO(2).

We now choose a basis, such that all basis vectors transform according to these irreducible representations. In this way we get a rotation matrix made up by a direct sum of irreducible representations. Any function h in the $\mathbb{L}^2$ function space can be expanded in that new basis as follows

$$h(x, y) = \sum_{m=-\infty}^{\infty} f_m(r) e^{im\varphi} \quad \text{with } \varphi = \arg(x + iy) \text{ and } r = \sqrt{x^2 + y^2}, \tag{3.20}$$

i.e. we express the function in polar coordinates, and make a *Fourier series* expansion of the function in angular coordinate φ, where m denotes the angular frequency. The functions $f_m(r)$ constitute the Fourier coefficients of the angular Fourier decomposition. Henceforth, we will indicate that a function is expressed in polar coordinates with a tilde, i.e. $\tilde{h}(r, \varphi) = h(r\cos\varphi, r\sin\varphi)$. Rotation of the function h can now simply be achieved by multiplying the Fourier components with a phase factor

$$\tilde{h}_\theta(r, \varphi) = \tilde{h}(r, \varphi - \theta) = \sum_{m=-\infty}^{\infty} e^{-im\theta} f_m(r) e^{im\varphi}. \tag{3.21}$$

Note that the radial functions $f_m(r)$ are not decomposed into a set of basis functions. This can be accomplished with the Hankel transform, but for the filter to be steerable, only the decomposition in angular coordinate is relevant.

3.5.2 Basics of steerable filters

In image processing, a filter is called *steerable* if the filter kernel has a finite number of angular frequency components, where the highest angular frequency is indicated by M [14]. In that case we can rewrite Equation 3.21 as

$$h_\theta(x,y) = \sum_{j=1}^{2M+1} k_j(\theta) h_j(x,y) = \mathbf{k}^{\mathrm{T}}(\theta)\mathbf{h}(x,y), \tag{3.22}$$

where $\mathbf{k}$ is a vector of $2M+1$ dimensions containing as coefficients functions of the desired rotation[3] (the term $e^{-im\theta}$ in Equation 3.21)

$$\mathbf{k}^{\mathrm{T}}(\theta) = \left(e^{iM\theta}, e^{i(M-1)\theta}, \ldots, 1, \ldots, e^{-i(M-1)\theta}, e^{-iM\theta}\right), \tag{3.23}$$

and $\mathbf{h}(x,y)$ is a vector of $2M+1$ dimensions containing the functions, in the form

$$\mathbf{h}(x,y) = \left(h_{-M}(x,y), h_{-M+1}(x,y), \ldots, h_0(x,y), \ldots, h_{M-1}(x,y), h_M(x,y)\right)^{\mathrm{T}}, \tag{3.24}$$

with

$$h_m(x,y) = f_m(\sqrt{x^2+y^2}) e^{im \arg(x+iy)}, \tag{3.25}$$

i.e. the term $f_m(r)e^{im\varphi}$ in Equation 3.21. We call the $h_m(x,y)$ functions the *m-components* that constitute the filter. If the filter kernel is real-valued, which is the case for our filters, then $f_m = f_{-m}$ holds for all $0 < |m| \leq M$.

The filter response of a filter in direction or orientation θ is obtained by calculating $r(\mathbf{b},\theta) = (L * h_\theta)(\mathbf{b})$. If we substitute Equation 3.22 in this equation we get

$$r(\mathbf{b},\theta) = (L * h_\theta)(\mathbf{b}) = \left\{ L * \left(\sum_{j=1}^{2M+1} k_j(\theta) h_j \right) \right\}(\mathbf{b}). \tag{3.26}$$

A convolution and a sum are both linear operations, which means they can be interchanged, giving

$$r(\mathbf{b},\theta) = \sum_{j=1}^{2M+1} k_j(\theta) \left(\{L * h_j\}(\mathbf{b})\right). \tag{3.27}$$

This equation says that if we first convolve the image with all functions h_j, we can calculate the filter response in any orientation, simply by a linear combination with coefficients $k_j(\theta)$. In this way, it is not necessary anymore to convolve the image with the filter kernel in all required orientations. Note, that if M is larger than the number of separate orientations that one wants to calculate, then simply convolving the image with the filter kernels in the desired orientations is probably more efficient.

The functions in Equation 3.25 directly relate to the polar Fourier series. All other valid basis filters can be constructed by linear combinations of the functions of Equation 3.25

$$\mathbf{h}'(x,y) = \mathrm{F}\,\mathbf{h}(x,y), \tag{3.28}$$

where F is a $(2M+1) \times (2M+1)$ matrix that must have an inverse F^{-1}. We call an arbitrary set of functions $\mathbf{h}'(x,y)$ the *basis filters* that constitute the filter. Equation 3.22 becomes

$$h_\theta(x,y) = (\mathbf{k}^{\mathrm{T}}(\theta)\mathrm{F}^{-1})(\mathrm{F}\,\mathbf{h}(x,y)), \tag{3.29}$$

so the new coefficients $\mathbf{k}'(\theta)$ are

$$\mathbf{k}'(\theta) = (\mathbf{k}^{\mathrm{T}}(\theta)\mathrm{F}^{-1})^{\mathrm{T}} = (\mathrm{F}^{-1})^{\mathrm{T}}\mathbf{k}(\theta). \tag{3.30}$$

We illustrate the design of a steerable filter with a simple example.

---

[3] Freeman et al. call this coefficients the *interpolation coefficients* but strictly speaking this is not correct terminology.

Example: the second order Gaussian derivative, case $\sigma_x = \sigma_y$

The isotropic second order Gaussian derivative filter kernel, in x-direction, is defined by

$$h(x,y) = G_\sigma^{(2,0)} = \frac{1}{2\pi\sigma^6} e^{-\frac{x^2+y^2}{2\sigma^2}} \left(x^2 - \sigma^2\right). \tag{3.31}$$

If we rewrite this in polar coordinates ($x \longrightarrow r\cos\varphi$, $y \longrightarrow r\sin\varphi$) we obtain using $\cos^2\varphi = \frac{1}{2}\cos 2\varphi + \frac{1}{2}$

$$\tilde{h}(r,\varphi) = \frac{1}{4\pi\sigma^6} e^{-\frac{r^2}{2\sigma^2}} \left(r^2 - 2\sigma^2 + r^2\cos(2\varphi)\right). \tag{3.32}$$

The kernel can now be rotated over an angle θ by substituting φ by $\varphi - \theta$

$$\tilde{h}_\theta(r,\varphi) = \tilde{h}(r,\varphi-\theta) = \frac{1}{4\pi\sigma^6} e^{-\frac{r^2}{2\sigma^2}} \left(r^2 - 2\sigma^2 + r^2\cos(2\varphi - 2\theta)\right). \tag{3.33}$$

To write the filter kernel in the form of Equation 3.22, we rewrite the trigonometric functions as exponential functions

$$\tilde{h}_\theta(r,\varphi) = \frac{1}{4\pi\sigma^6} e^{-\frac{r^2}{2\sigma^2}} \left(r^2 - 2\sigma^2 + \frac{1}{2}r^2\left(e^{2i\theta}e^{-2i\varphi} + e^{-2i\theta}e^{2i\varphi}\right)\right). \tag{3.34}$$

This can be written as

$$\tilde{h}_\theta(r,\varphi) = \mathbf{k}^{\mathrm{T}}(\theta)\mathbf{h}(r,\varphi) = \begin{pmatrix} e^{-2i\theta} \\ 1 \\ e^{2i\theta} \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} \frac{1}{8\pi\sigma^6} e^{-\frac{r^2}{2\sigma^2}} r^2 e^{2i\varphi} \\ \frac{1}{4\pi\sigma^6} e^{-\frac{r^2}{2\sigma^2}} (r^2 - 2\sigma^2) \\ \frac{1}{8\pi\sigma^6} e^{-\frac{r^2}{2\sigma^2}} r^2 e^{-2i\varphi} \end{pmatrix}, \tag{3.35}$$

so the filter contains an $m = -2$, $m = 0$ and $m = 2$ component, see Figure 3.7. If we now rewrite the functions $\mathbf{h}$ in cartesian coordinates, we can use Equation 3.27 to steer the filter to any direction using three basis filters.

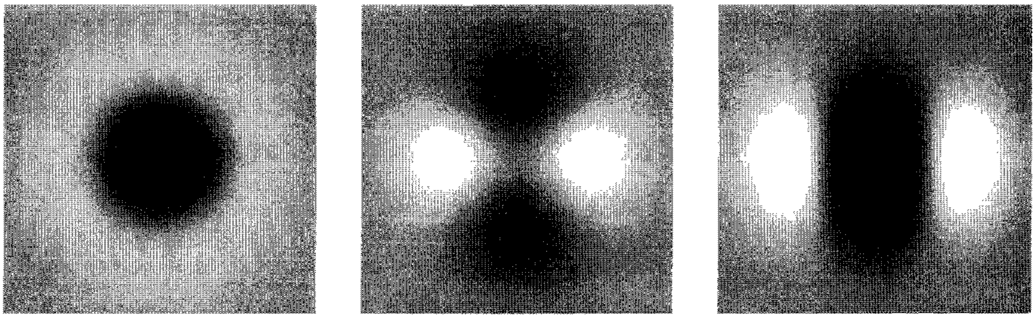

(a) (b) (c)

Figure 3.7: *The second order Gaussian derivative (c) consists of a $m = 0$ part (a) and a $|m| = 2$ part (b).*

Using Equation 3.29 we can construct a different basis, for instance, using $\mathrm{F} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ -i & 0 & i \end{pmatrix}$

$$\tilde{h}_\theta(r,\varphi) = \begin{pmatrix} 1 \\ \cos 2\theta \\ \sin 2\theta \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} \frac{1}{4\pi\sigma^6} e^{-\frac{r^2}{2\sigma^2}} (r^2 - 2\sigma^2) \\ \frac{1}{4\pi\sigma^6} e^{-\frac{r^2}{2\sigma^2}} r^2 \cos 2\varphi \\ \frac{1}{4\pi\sigma^6} e^{-\frac{r^2}{2\sigma^2}} r^2 \sin 2\varphi \end{pmatrix}. \tag{3.36}$$

Note that this is in fact the same as the decomposition of Equation 3.35 but with real-valued basis functions. This is more convenient for an implementation. An even more convenient, and most common, decomposition for steering the second order Gaussian derivative is (using $\mathrm{F} = \begin{pmatrix} 1 & 1 & 1 \\ -i & 0 & i \\ -1 & 1 & -1 \end{pmatrix}$)

$$\tilde{h}_\theta(r,\varphi) = \begin{pmatrix} \frac{1}{2}(1+\cos 2\theta) \\ \sin 2\theta \\ \frac{1}{2}(1-\cos 2\theta) \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} G_\sigma^{(2,0)} \\ G_\sigma^{(1,1)} \\ G_\sigma^{(0,2)} \end{pmatrix}, \tag{3.37}$$

which has the advantage of consisting of x, y-separable basis functions (i.e. we can write $h(x,y) = f(x)g(y)$ for all basis filters).

Example: the second order Gaussian derivative, case $\sigma_x \neq \sigma_y$

The anisotropic 2D second order Gaussian derivative filter kernel in x-direction, with two different scales σ_x and σ_y is defined by

$$h(x,y) = G^{(2)}_{\sigma_x}(x) G_{\sigma_y}(y) = \frac{1}{2\pi\sigma_x^5\sigma_y} e^{-\frac{x^2}{2\sigma_x^2}-\frac{y^2}{2\sigma_y^2}} \left(x^2 - \sigma_x^2\right). \tag{3.38}$$

If we rewrite this to the polar coordinate system and rewrite the trigonometric function to exponential functions we obtain

$$\tilde{h}_\theta(r,\varphi) = \frac{1}{4\pi\sigma_x^5\sigma_y} e^{-\frac{1}{2}r^2\left(\left(\frac{e^{-i\varphi}+e^{i\varphi}}{2\sigma_x}\right)^2-\left(\frac{e^{-i\varphi}-e^{i\varphi}}{2\sigma_y}\right)^2\right)} \left(r^2 - 2\sigma_x^2 + \frac{1}{2}r^2\left(e^{-2i\varphi}+e^{2i\varphi}\right)\right). \tag{3.39}$$

Note the φ-dependency within the exponential term! This dependency makes it impossible to make this kernel steerable with a limited number of components. The only possibility would be to create a steerable filter that *approximates* this filter, for instance using a Taylor expansion of the exponential.

3.5.3 Choice of radial functions

We want to find a suitable basis for constructing steerable ridge detection filters. We have seen that the only suitable basis for the angular part of the filter is already determined. These functions must be of the form $e^{im\varphi}$, $m \in \mathbb{Z}$. A basis for the radial functions $f_m(r)$, however, can be freely chosen.

To form a basis that spans the entire $\mathbb{L}^2$ function space, we would also need to decompose the radial functions into a set of basis functions. A possible choice of basis functions used in the *Hankel transformation* is

$$H^n_m(z) = \sqrt{\frac{2n!}{(n+|m|)!}}\, z^m e^{-\frac{|z|^2}{2}} L^{|m|}_n(|z|^2), \tag{3.40}$$

where $z = x + iy$ (cartesian coordinates) or $z = re^{i\varphi}$ (polar coordinates), and $L^{|m|}_n$ denotes the generalized Laguerre polynomials. These functions are the eigenfunctions of the Fourier transform. To make the filter steerable it is necessary to truncate the expansion at a finite value of m.

For the design of our ridge filters, we choose not to do an expansion in n, but instead to take only a single basis function for every value of m. We choose for our basis functions g_m the Fourier eigenfunctions with $n = 0$ for $m \neq 0$. For $m = 0$ we make an exception, in order to obey the requirement of invariance under grayscale translation (see Subsection 3.4.2), we choose to use a linear combination of H^0_0 and H^1_0 that does fulfill this requirement. So our basis is defined as

$$\tilde{g}_m(r,\varphi;\sigma) = \begin{cases} \frac{\sqrt{2^{1-m}m!}}{4\pi\sigma^2} H^0_m\left(\frac{re^{i\varphi}}{\sigma}\right) = \frac{1}{2\pi\sigma^2}\left(\frac{r}{\sqrt{2}\,\sigma}\right)^{|m|} e^{-\frac{r^2}{2\sigma^2}} e^{im\varphi} & \text{for } m \neq 0, \\ -\frac{1}{8\pi\sigma^2}\left\{H^0_0\left(\frac{re^{i\varphi}}{\sigma}\right) + H^1_0\left(\frac{re^{i\varphi}}{\sigma}\right)\right\} = \frac{1}{4\sqrt{2}\,\pi\sigma^4} e^{-\frac{r^2}{2\sigma^2}}\left(r^2 - 2\sigma^2\right) & \text{for } m = 0, \end{cases} \tag{3.41}$$

see Figure 3.8. In cartesian coordinates this gives

$$g_m(x,y;\sigma) = \begin{cases} \frac{1}{2\pi\sigma^2}\frac{1}{(\sqrt{2}\,\sigma)^{|m|}}\, e^{-\frac{x^2+y^2}{2\,\sigma^2}}\,(x+iy)^{|m|} & \text{for } m > 0, \\ \frac{1}{2\pi\sigma^2}\frac{1}{(\sqrt{2}\,\sigma)^{|m|}}\, e^{-\frac{x^2+y^2}{2\,\sigma^2}}\,(x-iy)^{|m|} & \text{for } m < 0, \\ \frac{1}{4\sqrt{2}\,\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}}\left(x^2+y^2-2\sigma^2\right) & \text{for } m = 0. \end{cases} \tag{3.42}$$

For convenience these definitions are chosen to be scale invariant. We call g_m the *Gaussian m functions.*

The ad hoc motivations for choosing these basis functions are:

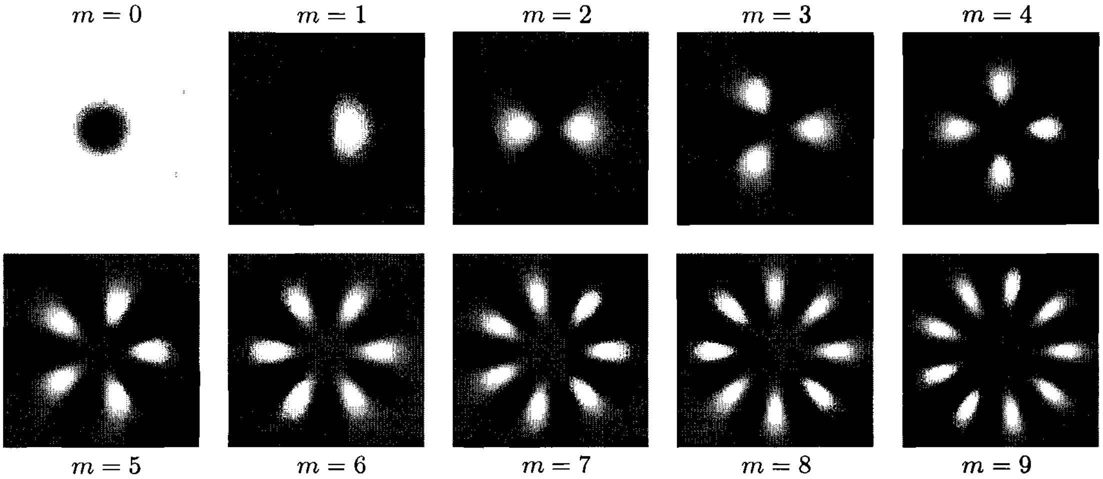


Figure 3.8: *The Gaussian m basis functions, from $m = 0$ to $m = 9$.*

- We want the filter to gradually decay when going to infinity. This is accomplished by the Gaussian window function $e^{-\frac{r^2}{2\sigma^2}}$.
- The radial part $r^{|m|}$ makes the basis functions more distant from the center for higher values of m. This prevents aliasing to occur in the center of the sampled basis functions. Also, this property makes this a quite natural basis to construct a ridge filter that becomes more elongated for increasing m.
- In [22], Kalitzin et al. use the same basis (for $m \neq 0$) to construct an *orientation score* of an image, from which the original image can be reconstructed, if sufficiently large values for m are taken into account. Inspired by this approach we started to work with the same basis. However, we observed that the main problem with the filters used by Kalitzin is that the filter does not have the highest value in the center, leading to some displacement when detecting ridges. Also, the filter is not invariant under grayscale translation. Therefore, we choose the Laplacian for the $m = 0$ basis function, based on the observation that the second order Gaussian derivative (which is a good ridge detector) consists of the Laplacian (the g_0 part) and a g_2 part.

The basis functions g_m can be constructed entirely from the Gaussian derivatives, as follows [22]

$$g_m(x,y;\sigma) = \begin{cases} \left(\frac{-\sigma}{\sqrt{2}}\right)^{|m|}\left(\frac{\partial}{\partial x}+i\frac{\partial}{\partial y}\right)^{|m|} G_\sigma(x,y) & \text{for } m>0,\\ \left(\frac{-\sigma}{\sqrt{2}}\right)^{|m|}\left(\frac{\partial}{\partial x}-i\frac{\partial}{\partial y}\right)^{|m|} G_\sigma(x,y) & \text{for } m<0,\\ \frac{\sigma^2}{2\sqrt{2}}\left(\left(\frac{\partial}{\partial x}\right)^2+\left(\frac{\partial}{\partial y}\right)^2\right) G_\sigma(x,y) & \text{for } m=0. \end{cases} \tag{3.43}$$

This makes it easy to construct the basis by calculating Gaussian derivatives of the image, which have the advantage of being x, y-separable.

3.5.4 Determining the optimal weighting coefficients

We want to create an optimal filter of the form

$$h(x,y) = \sum_m \alpha_m g_m(x,y), \tag{3.44}$$

for detecting catheters. The coefficients α_m are the weighting factors of the different m-components. In this subsection, we will find values for these coefficients, based on the catheter profile function of section Subsection 3.4.1. This is achieved by projecting the catheter profile function $l_{\varrho,\sigma_d}(x,y)$, further on abbreviated as $l(x,y)$, onto the basis spanned by the Gaussian m functions g_m. These basis functions are orthogonal, with the following orthogonality relation

$$\langle g_{i,\sigma} \mid g_{j,\sigma} \rangle = \delta_{ij} \frac{2^{-i} i!}{4\pi\sigma^2}, \tag{3.45}$$

where $g_{i,\sigma}$ denotes the basis function g_i at scale σ, the $\mathbb{L}^2$ inner product is defined as in Equation 3.2, and

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j; \\ 0 & \text{if } i \neq j. \end{cases} \tag{3.46}$$

We can project the function $l(x,y)$ onto these basis vectors as follows

$$\mathbb{P}_g(l) = \sum_m \frac{\langle g_{m,\sigma} \mid l \rangle}{\langle g_{m,\sigma} \mid g_{m,\sigma} \rangle} g_{m,\sigma}. \tag{3.47}$$

So optimal values for the coefficients α_m can be calculated by

$$\alpha_m = \frac{\langle g_{m,\sigma} \mid l \rangle}{\langle g_{m,\sigma} \mid g_{m,\sigma} \rangle}. \tag{3.48}$$

Our catheter model $l(x,y)$ is real-valued, i.e. $\alpha_m = \overline{\alpha_{-m}}$, so it is sufficient to only do the projection for $0 \leq m \leq M$ where M is the highest m-component in the filter. To indicate that the $+m$ and $-m$ components are combined, we write

$$\alpha_{|m|} = 2\alpha_m = 2\overline{\alpha_{-m}} \quad \text{for } m > 0. \tag{3.49}$$

Since our catheter model has a 180° symmetry, we only take *even* m-components into account.

We have a free scale parameter σ. The optimal values for α_m should be calculated at the optimal scale σ_{optimal}. We, thus, need an optimal scale selection mechanism. We consider the optimal scale as the scale, at which the $\mathbb{L}^2$-norm of the projection to the space spanned by the *orthonormal* g_m functions is maximal. The orthonormal projection onto the g_m functions is given by

$$\alpha'_m = \frac{\langle g_{m,\sigma} \mid l \rangle}{\sqrt{\langle g_{m,\sigma} \mid g_{m,\sigma} \rangle}}, \tag{3.50}$$

so the norm of the vector $\boldsymbol{\alpha}' = (\alpha'_0, \alpha'_1, \alpha'_2, \ldots)$ is given by

$$||\boldsymbol{\alpha}'||_{\mathbb{L}^2} = \sum_{m=-M}^{M} \frac{|\langle g_{m,\sigma} \mid l \rangle|^2}{\langle g_{m,\sigma} \mid g_{m,\sigma} \rangle} = \frac{|\langle g_{0,\sigma} \mid l \rangle|^2}{\langle g_{0,\sigma} \mid g_{0,\sigma} \rangle} + 2 \sum_{m=1}^{M} \frac{|\langle g_{m,\sigma} \mid l \rangle|^2}{\langle g_{m,\sigma} \mid g_{m,\sigma} \rangle}. \tag{3.51}$$

To be able to compare these norms over different scales, we need to make this expression invariant with respect to scale. In [29], Lindeberg shows that for Gaussian derivatives this can be achieved by introducing dimensionless coordinates $\xi = x/\sigma$, resulting in normalized derivative operators $\left(\frac{\partial f}{\partial \xi}\right)^n = \sigma^n \left(\frac{\partial f}{\partial x}\right)^n$. In Equation 3.43 it can be seen that our definitions for the basis functions g_m already comply with this normalized derivative operator, so the term $|\langle g_{m,\sigma} \mid l \rangle|^2$ is already normalized with respect to scale. The term $\langle g_{m,\sigma} \mid g_{m,\sigma} \rangle$, however, introduces a $1/\sigma^2$ term, which should be compensated to keep invariance with respect to scale. So, the optimal scale can be found by

$$\sigma_{\text{optimal}} = \arg\max_{\sigma} \frac{1}{\sigma^2} \sum_{m=-M}^{M} \frac{|\langle g_{m,\sigma} \mid l \rangle|^2}{\langle g_{m,\sigma} \mid g_{m,\sigma} \rangle}. \tag{3.52}$$

The catheter profile function has two parameters: catheter radius r and device transfer function scale σ_{d}. For the calculation of the filter coefficients α_m, the ratio between these two parameters is sufficient to know. We roughly estimated $\varrho \approx 3.5$ pixels, and using measurements in [26] we estimated $\sigma_{\text{d}} \approx 0.64$ pixels for 512×512 image resolution, so $\sigma_{\text{d}}/r \approx 0.18$. The resulting coefficients are not very sensitive to the exact ratio.

Table 3.1 shows the resulting coefficients α_m for different values of M. Figure 3.9 shows the resulting filters for different values of M. Note that the resulting filter for $M = 2$, which has $\alpha_0 = 1$ and $\alpha_{|2|} = 2$, is not the same as the second order Gaussian derivative filter, which has $\alpha_0 = 1$ and $\alpha_{|2|} = 1$ (compare Figure 3.9a-b). Apparently, according to our method the $m = 2$ should be weighted stronger. In [33], Meijering et al. derive exactly the same filter, using a flatness criterion. For the $M = 4$ and $M = 6$ filter, artefacts can be observed in the shape of the side lobs (the white parts) of the filter, due to the limited amount of basis functions that are used to construct the filter. It is not clear yet whether these artefacts are disturbing for the ridgeness result.

M	rel. σ_{opt}	α_0	$\alpha_{\vert 2\vert}$	$\alpha_{\vert 4\vert}$	$\alpha_{\vert 6\vert}$	$\alpha_{\vert 8\vert}$
2	1	1	2			
4	1.09	1	2	-0.72		
6	1.17	1	2	-0.74	0.18	
8	1.23	1	2	-0.74	0.18	-0.030

Table 3.1: *Weighting coefficients α_m for different values of M. M indicates the highest m-component that is taken into account. The values for α_m are scaled such that α_0 is one. The column "rel. σ_{opt}" gives the optimal scale relative to the optimal scale for $M = 2$, showing that the optimal scale increase when higher m-components are taken into account.*

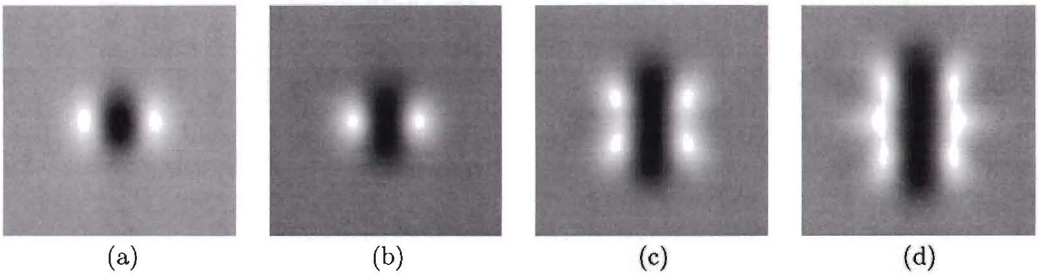

(a) (b) (c) (d)

Figure 3.9: *(a) "Normal" second order Gaussian derivative. (b) $M = 2$ filter, same as the filter proposed by Meijering et al. (c) $M = 4$ filter. (d) $M = 6$ filter.*

3.5.5 Influence of noise

Now that we are able to construct steerable ridge detection filters that become more elongated for higher values of M, the following question arises: what m-components are sensible to take into account?

On one hand, one can argue that including higher values of m lead to better noise robustness and more precise orientation information, because a larger surrounding is taken into account by a more elongated filter. In Figure 3.10, an example is shown, in which the $M = 4$ filter gives better orientations on the catheter than the $M = 2$ filter.

On the other hand, one can argue that higher m-components contain higher frequencies, resulting in a less smooth filter that is more sensitive to noise. Empirically we can determine the usefulness of the different angular frequency components by inspecting the result of convolving an image with a single m-component as defined in Equation 3.41. If the structures of interest are still "clearly" visible in the resulting image, there must be useful information in it, otherwise not.

Figure 3.11 shows the first four, even m-components in an X-ray image with two noise levels. The scale of the filter is set to $\sigma = 3.4$ (will be determined in Chapter 6, Subsection 6.1.3). It can be clearly seen that for the high-quality image, useful information is contained in all m-components. In the low-quality image, only the $m = 0$ and $|m| = 2$ components visually contain useful information.

Based on these observations, we conclude that taking high m-components into account is not likely to be effective for the noisy images. Another argument to limit the number of m-components

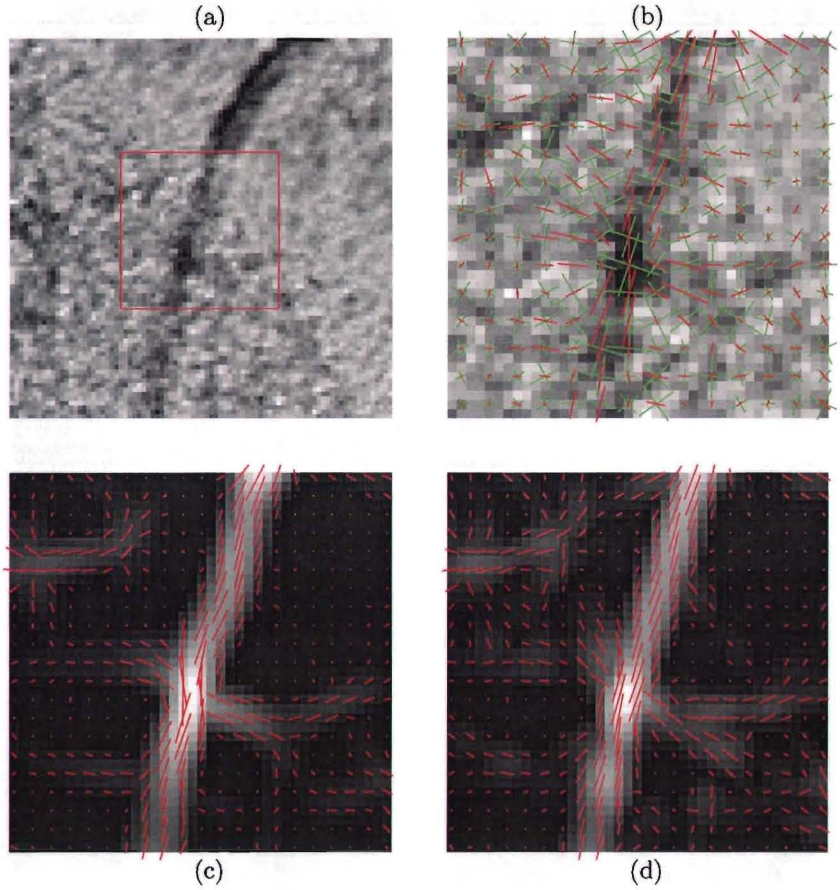


Figure 3.10: *Illustration of directional fields of an X-ray image. (a) Original background equalized image. The rectangle shows the zoom area for the next figures. (b) Directional fields of $|m| = 2$ and $|m| = 4$ components. The red lines show the length and complex phase of the $|m| = 2$ component. The green crosses show the length and complex phase of the $|m| = 4$ component (which has a 90°-symmetry). (c) Result of the $M = 2$ filter. The superimposed red lines show the strength and orientation of the filter. (d) Result of the $M = 4$ filter. The superimposed red lines show the strength and orientation of the filter. Note the difference in the center of the image: the $M = 4$ filter seems to give better orientations.*

is the larger computational demand. Henceforth, we will only consider the filters with $M = 2$ and $M = 4$.

3.5.6 Implementation of steerable filters

We want to find the maximal response of the steerable filter with respect to the orientation, see Equation 3.17. Figure 3.12 shows the way a steerable filter is implemented. First, the Gaussian derivatives are calculated. From these derivatives the different m-components are calculated. Then, for every pixel position the filter is aligned with the orientation of maximum response. The result is a local ridgeness feature image r and an orientation image ψ, where $\psi(x, y)$ is defined as the measured orientation *tangent* to the local ridge. An example of the results of both filters is shown in Figure 3.13.

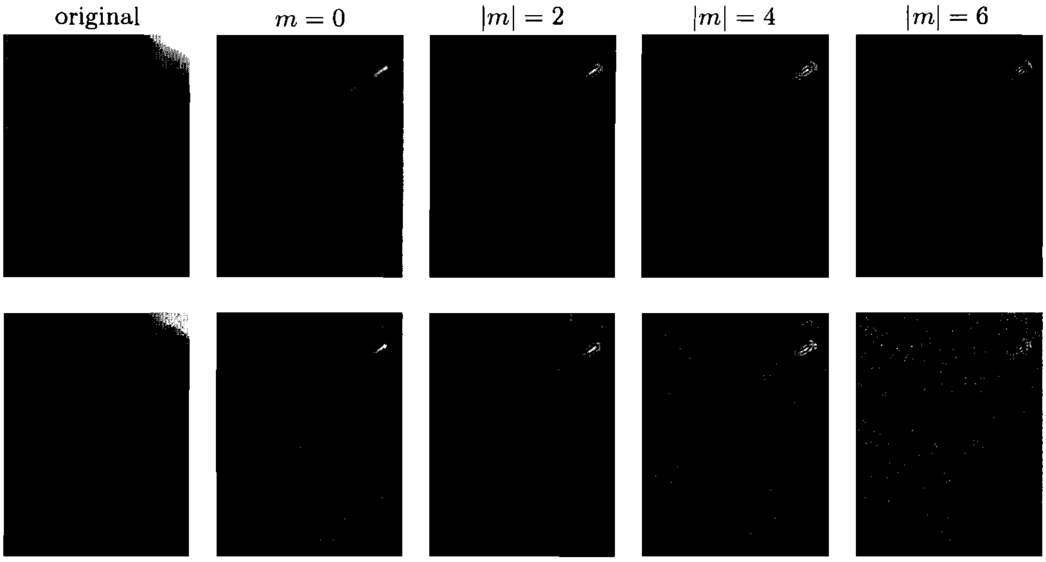


Figure 3.11: *Effect of different g_m filter components on an X-ray image. The top row shows the results on an image with low noise. The bottom row shows the results for the same image but with more noise. For $|m| = 2$, $|m| = 4$, and $|m| = 6$ the absolute values of the responses are shown. Notice that for the low noise image, the catheter is visible in all images, so all components are likely to contain useful information. For the image with high noise, however, the catheter is hardly visible in the $|m| = 4$ and $|m| = 6$ image.*

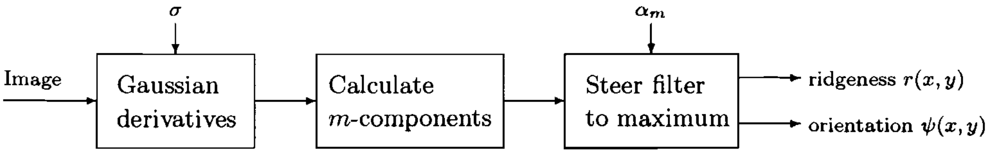


Figure 3.12: *Implementation of a steerable filter.*

Implementation of $M = 2$ filter

The m-components of the $M = 2$ filter can be calculated by

$$u_0(x, y) = L_\sigma^{(2,0)}(x, y) + L_\sigma^{(0,2)}(x, y), \tag{3.53}$$

$$u_2(x, y) = L_\sigma^{(2,0)}(x, y) - L_\sigma^{(0,2)}(x, y) + 2i\, L_\sigma^{(1,1)}(x, y), \tag{3.54}$$

where $L_\sigma^{(i,j)}(x, y)$ is the i, j-th derivative of image L at scale σ. The ridgeness in an arbitrary orientation θ is given by

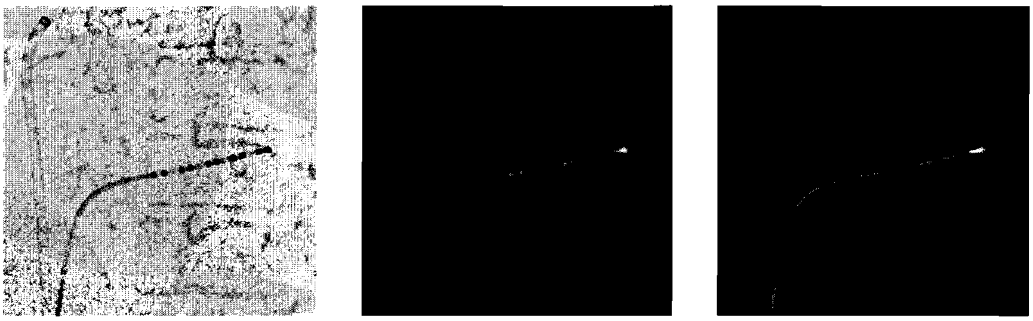
$$r(\mathbf{b}, \theta) = \mathrm{Re}\{\alpha_0\, u_0(\mathbf{b}) + \alpha_{|2|}\, e^{-i2\theta}\, u_2(\mathbf{b})\}, \tag{3.55}$$

where $\alpha_0 = 1$ and $\alpha_{|2|} = 2$ (Table 3.1). From this it can be seen that the maximum ridgeness response and orientation are given by

$$r(x, y) = \max\left(\alpha_0 u_0(x, y) + \alpha_{|2|} |u_2(x, y)|, 0\right), \tag{3.56}$$

$$\psi(x, y) = \frac{1}{2} \arg\left(-u_2(x, y)\right). \tag{3.57}$$

The negative values for the ridgeness r are truncated to zero, because we are searching for dark ridges relative to the background. These ridges should always give a positive response. Note the minus sign in the calculation of ψ. This is necessary to get the orientation *tangent* to the ridge instead of perpendicular to the ridge.



(a) (b) (c)

Figure 3.13: *Result of different ridge detectors. (a) Background equalized input image. (b) Result of the $M = 2$ filter. (c) Result of the $M = 4$ filter. Note the difference in responses. The response on the catheter seems to be more consistent for the $M = 4$ filter. However, the area surrounding the catheter is more messy for the $M = 4$ filter.*

Note that exactly the same results are obtained by calculating the eigenvalues and eigenvectors of the Hessian matrix $\mathrm{H} = \begin{pmatrix} L_\sigma^{(2,0)} & L_\sigma^{(1,1)} \\ L_\sigma^{(1,1)} & L_\sigma^{(0,2)} \end{pmatrix}$ [4].

Implementation of $M = 4$ filter

The $M = 4$ filter has the additional $|m| = 4$ component, which is

$$u_4(x,y) = \sigma^2 \left(L_\sigma^{(4,0)}(x,y) - 6L_\sigma^{(2,2)}(x,y) + L_\sigma^{(0,4)}(x,y) + 4i \left(L_\sigma^{(3,1)}(x,y) - L_\sigma^{(1,3)}(x,y) \right) \right). \tag{3.58}$$

The total filter response in orientation θ is calculated by

$$r(\mathbf{b}, \theta) = \mathrm{Re}\left(\alpha_0 \, u_0 + \alpha_{|2|} \, e^{-i2\theta} u_2 + \alpha_{|4|} \, e^{-i4\theta} u_4 \right), \tag{3.59}$$

where the values of α_m are determined in Subsection 3.5.4, see Table 3.1. The filter is steered in the direction, in which the response is maximal by solving for every pixel coordinate $(x, y) \in \Omega$

$$r(x,y) = \max \left(\max_\theta \mathrm{Re}\left(\alpha_0 \, u_0(x,y) - \alpha_{|2|} \, e^{-i2\theta} u_2(x,y) + \alpha_{|4|} \, e^{-i4\theta} u_4(x,y) \right), 0 \right), \tag{3.60}$$

$$\psi(x,y) = \arg\max_\theta \mathrm{Re}\left(\alpha_0 \, u_0(x,y) - \alpha_{|2|} \, e^{-i2\theta} u_2(x,y) + \alpha_{|4|} \, e^{-i4\theta} u_4(x,y) \right). \tag{3.61}$$

Similar to the $M = 2$ filter, negative values are truncated, because we are searching for dark ridges. The minus sign in front of the $\alpha_{|2|}$ is again necessary to get the orientation tangent to the ridge. Solving this equation analytically leads to a rather difficult expression. In the implementation, therefore, we solve this numerically by first estimating the maximum, and then using the Newton-Rhapson method to iterate to the right solution. The algorithm is as follows

---

[4] This is derived in the next chapter in Subsection 4.2.1

Algorithm 1: *Steer the $M = 4$ filter to the orientation with maximum response*

```
For all pixels $\mathbf{b} \in \Omega$ do:
  • Calculate complex phases of $m = 2$ and $m = 4$ components, and unwrap phase of $m = 4$ component:
    $\phi_2 = \frac{1}{2} \arg u_2(\mathbf{b})$
    $\phi_4 = \frac{1}{2} \arg u_4(\mathbf{b})$
    If $(\phi_4 - \phi_2 \geq \pi/4)$ then $\phi_4 = \phi_4 - \pi/2$
    If $(\phi_4 - \phi_2 \leq -\pi/4)$ then $\phi_4 = \phi_4 + \pi/2$
  • Use $\phi_2$ and $\phi_4$ for an initial estimation:
    $\phi_{\text{estim}} = \arg\max_{\phi} \{r(\mathbf{b}, \phi_2), r(\mathbf{b}, \phi_4), r(\mathbf{b}, \frac{\phi_2+\phi_4}{2})\}$
  • Use Newton-Rhapson method to iterate to best orientation
    $i = 0$
    Do
      $\Delta = \left(\frac{\partial}{\partial \phi} r(\mathbf{b}, \phi_{\text{estim}})\right) / \left(\frac{\partial^2}{\partial \phi^2} r(\mathbf{b}, \phi_{\text{estim}})\right)$
      $\phi_{\text{estim}} = \phi_{\text{estim}} - \Delta$
      $i = i + 1$
    While ($|\Delta| \leq$ NR_STOP_THRESHOLD and $i \leq$ NR_MAX_ITERATIONS )
  • Return orientation $\phi_{\text{estim}}$ and response $\max(r(\mathbf{b}, \phi_{\text{estim}}), 0)$
```

There are only few (max. 5) iterations required to get to a precise solution (i.e. error smaller than 10^{-9} rad). The calculation of the $M = 4$ filter is computationally more expensive than the $M = 2$ filter. Nevertheless, it is still more efficient than just calculating ridgeness in, for instance, 6 different orientations by convolving the image with a filter kernel in 6 different orientations: Pixel-wise operations that are needed to steer a steerable filter are cheaper than convolutions, and render more precise orientations.

3.6 Blob detection

The electrodes on the EP catheter are a useful cue for detecting the catheters with higher certainty. Electrodes on the EP catheters are situated on a line structure, and are nearly isotropic (except for the ablation tip on ablation catheters).

The second eigenvalue of the Hessian

$$b(x, y) = \max\left(0\,,\ \frac{1}{2}\left(L_{xx} + L_{yy} - \sqrt{4\,L_{xy}^2 + (L_{xx} - L_{yy})^2}\right)\right) = \max\left(0\,,\ \frac{1}{2}(u_0 - |u_2|)\right), \quad (3.62)$$

proves to be a good electrode detector. If the second eigenvalue λ_2 is large and positive, the first eigenvalue λ_1 is also large and positive (because $\lambda_1 \geq \lambda_2$). The eigenvalues of the Hessian are the principal curvatures of the intensity landscape, so a large λ_2 indicates a *concave* shape in the intensity profile. This matches well with electrodes on EP catheters, which appear as dark blobs in the image.

The Hessian blob detector has the advantage that the second order Gaussian derivatives are already calculated for the ridge detection. Another advantage is that the Hessian blob detector also gives a high response on both end-points of an ablation tip, so we can also detect ablation tips with this detector. Figure 3.13 shows an example of the blob detection result.

3.7 Notes on scale

A very important parameter for local feature detection is scale. For this application, we have to add a few remarks on scale.

With fixed settings of the X-ray device, the catheter will have approximately the same scale everywhere. The catheter has a constant physical thickness, and the effect of varying distance to the diverging X-ray beam is expected to be minor. Therefore, a full multi-scale analysis does not have much added value once the appropriate scale is known.

However, for different settings of the X-ray system the optimal scale can vary a lot, especially for varying distances from the X-ray beam to the detector. A scale selection mechanism might be

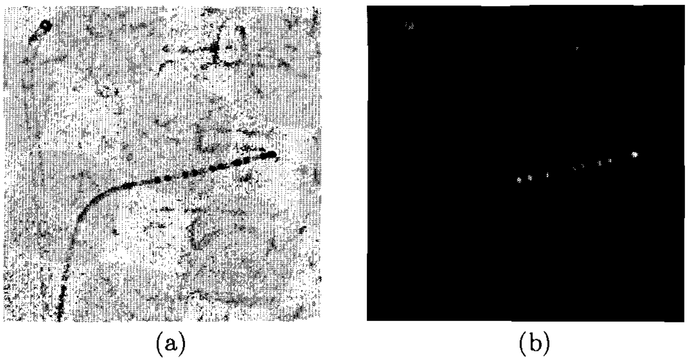


Figure 3.14: *Blob detection example. (a) Background equalized image. (b) Blob detection with λ_2 of the Hessian.*

needed. The classical scale selection method using normalized derivatives proposed by Lindenberg [29] can be used for this purpose, see also Subsection 3.5.4.

Before the scale can be measured, it is necessary to know the position of the catheter in the image. There are two possibilities. Either the user points to the catheter manually for the first frame and the scale is measured on this manual pointed position, or the algorithm first tries to detect the catheter on a scale that is typical for a catheter in "normal clinical" situations[5]. Because the scale parameter is not extremely sensitive, this will often work well. If the detected catheter is very likely to be correct (for example, by asking the user if the extraction is correct, or by detecting the catheter in several frames and checking the consistency), a scale measurement can be performed. In Section 6.1 the optimal scale will be measured on a few manually segmented images.

3.8 Conclusions

In this chapter we introduced methods to detect local features in EP catheter images. We paid a lot of attention to steerable filters, which is a very useful concept regarding efficiency. We will need the knowledge on steerable filters in the next chapter, where we will use steerable filter theory to derive an efficient context enhancement scheme.

Based on a model of a straight catheter, we derived optimal filters in a steerable basis, by projecting the model onto this basis with optimal chosen scale. The resulting filters are similar to the common second order Gaussian derivative, but are more elongated for higher M. It is expected that more elongated filters will lead to better results, but this should be experimentally verified. This will be the topic of Chapter 6, where we will compare the $M = 2$, $M = 4$, and the normal second order Gaussian derivative filter.

After finishing this work, we found an interesting new article by Jacob et al. [20]. In this article, they also try to obtain better ridge detectors by making more elongated steerable filters based on Gaussian derivatives. Interesting in this article is the optimality criterion they use, which includes optimization with respect to noise-robustness, localization error, and elimination of false oscillations. They also propose a way to analytically find the best orientation (the orientation with max. response) for the $M = 4$ filter. This article is worth looking at for further research.

The filters described in this chapter still only use local spatial information. In the next chapter, the output of these local filters will be communicated over a larger neighborhood.

---

[5] Because we have a limited number of clinical images available, it is unknown how much variance there is in X-ray system settings between different EP procedures. This should be investigated in more detail.

Chapter 4

Context Enhancement

In the previous chapter, we introduced filters for detecting local features in EP catheter images. For very noisy images the information obtained using these local filters is often poor. Gaps in line structures, for instance, make the next step, the extraction of the EP catheters, difficult. The aim of *context enhancement* is to improve local feature data using knowledge of the spatial neighborhood. The idea is to let local image sites *communicate* with each other over a larger neighborhood (the context) to obtain spatially consistent features.

Figure 4.1 shows the "black-box" scheme of the context enhancement part. As input, the local features of the previous chapter are needed. As output, context enhanced feature measures are obtained.

The structure of this chapter is as follows. In Section 4.1 we will supply some backgrounds to explain the approach. In Section 4.2 we will describe the *tensor voting* technique. This is a context enhancement technique described in literature [32] that is an important source of inspiration for this work. In Section 4.3, we modify the tensor voting method with the use of steerable filters, as introduced in Chapter 3, resulting in a more efficient computational scheme. In Section 4.4, the application of the proposed techniques on EP catheter images is described in detail. Finally, conclusions are drawn.

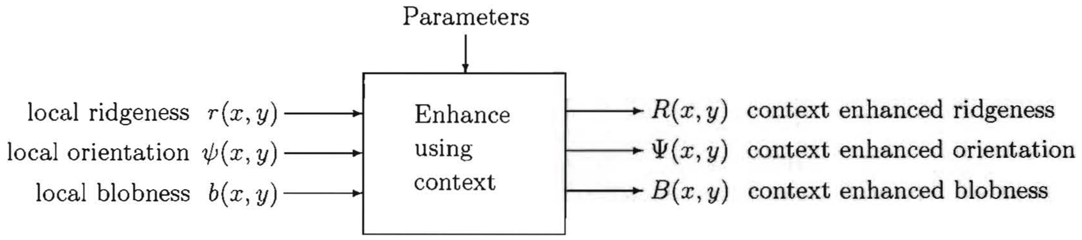
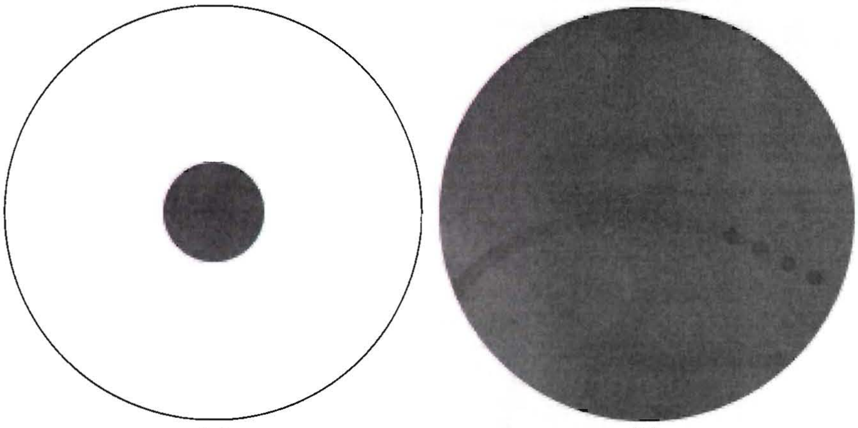


Figure 4.1: *Black-box of context enhancement. As input, the results of the local feature detectors of Chapter 3 are used. As output, context enhanced feature images are generated, which are used in Chapter 5 for EP catheter extraction.*

4.1 Background

Figure 4.2 illustrates the importance of spatial context. If we look through a small window as a local feature detector does, structures are hardly visible. If we see a larger part of an image, suddenly the structure of interest emerges. Apparently, for our visual system the spatial context is an important clue for object detection.



(a) (b)

Figure 4.2: *EP catheter example, demonstrating of the importance of context. (a) Viewing the image through a small window. The small part of the image contains an EP catheter, but the catheter is hardly visible. This is the way a local filter looks at the image. (b) If we enlarge the window, we can clearly see the EP catheter. Apparently, the visual system uses spatial context information to enhance the detection of features.*

4.1.1 Biological motivation

Recent findings about the way the biologic visual system works are an important source of inspiration, see e.g. [19, 23] for an introduction to this topic. Here, only the most important findings for us are summarized shortly.

In the retina, Laplacian-like filters can be found at various scales, which suggests the need for a *multi-scale* image analysis of the *differential structure* of images [37]. In the primary visual cortex (PVC), cells are found that respond to specific orientations. This implies that the use of *orientation selective filters* is essential. The methods we described in Chapter 3 nicely link to these biological findings.

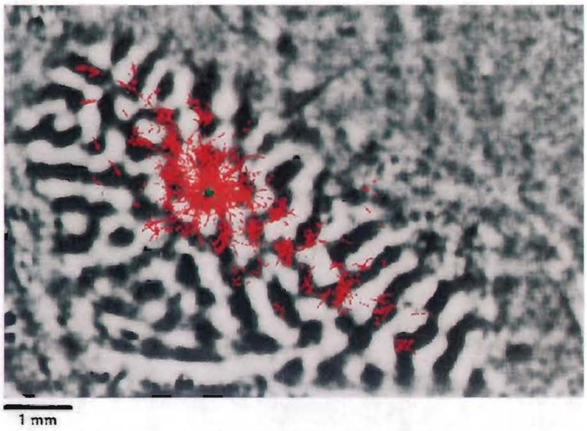


Figure 4.3: *Activation patterns in the primary visual cortex. The gray value indicates the orientation selectivity of cells in the different areas. The black areas respond to line structures that are orthogonal to the line structures the white areas respond to. The green symbols indicate cells that took up and transported biocytin, a substance that visualizes connections between cells. The red symbols indicate locations of cells that communicated with the cells indicated by the green symbols. From [5].*

Recently, it has been found that these orientation selective cells have connections with neighboring cells with approximately the same orientations [5]. This neighborhood of a cell covers a fairly large part of the PVC (Figure 4.3). This communication between cells is probably an important mechanism for the perception of elongated structures. It suggests the use of *contextual filters* that emulate communication of image sites over a larger neighborhood.

4.1.2 Perceptual grouping and the Gestalt principles

The problem of detecting EP catheter is a *perceptual grouping* problem. Perceptual grouping is a commonly used name for problems in image analysis, where low level image primitives (e.g. local evidence of ridges, local evidence of blobs) are *grouped* to *higher level primitives*, e.g. the *global objects* of interest that appear in the image. This grouping process is an important step for the interpretation of images.

An example is given in figure Figure 4.4. Small line segments are grouped together to form longer line segments. Of course there does not exist one unique solution to this problem. It is very hard to make a general applicable perceptual grouping algorithm. In practice it is necessary to use application-specific knowledge.

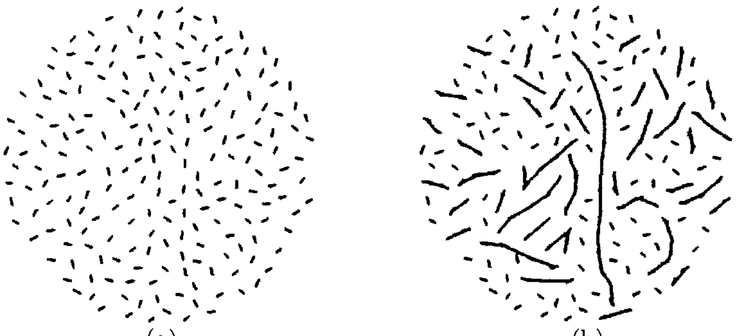


Figure 4.4: *An example of the perceptual grouping problem: how to group the small local line segments (a) into global structures (b). Source:* http://www.cps.utexas.edu/Research/Geisler/Projects/perceptualgrouping.html.

To group objects, we need to define *grouping rules*. The work of the Gestalt psychologists forms an important source of inspiration. The Gestalt psychologists, at the beginning of the 20th century formulated qualitative grouping rules, which are very likely to be used in the human visual system (Figure 4.5):

1. *Proximity*: segments that are close together are more likely to belong to the same object.
2. *Good continuation*: elements are inclined to be grouped such that the results have smooth and continuous characteristics. In other words, small curvature is favored over large curvature.
3. *Closure*: there is a tendency for curves to be completed so that they form enclosed regions.
4. *Similarity*: similar elements are likely to belong to the same object.
5. *Common Fate*: elements that appear to move together belong to the same object.
6. *Symmetry*: the structure of a figure or object tends to have a symmetry.
7. *Figure-ground segregation*: for an object to be perceived, it must stand apart from its background.

Note that almost all these rules are applicable on the EP catheter detection problem. For our problem, we can of course define more specific perceptual grouping rules, e.g. electrodes are situated on EP catheters, with a specific spacing, etcetera. The model used for the context enhancement in this chapter is still quite general. The application-specific grouping for EP catheters will be subject of the next chapter (Chapter 5).

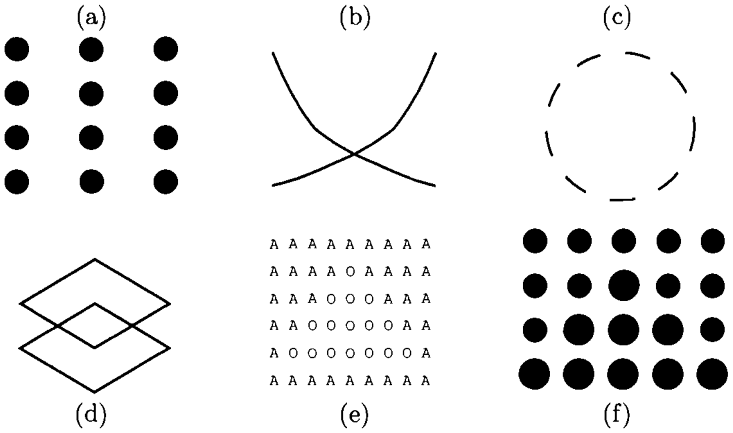


Figure 4.5: *Examples of Gestalt principles. (a) proximity, (b) good continuation, (c) closure, (d) symmetry, (e) similarity (shape), and (f) similarity (size).*

4.2 Review of tensor voting

Tensor voting (abbreviated as "TV") is a method to segment and group structures like curves, junctions, regions, and surfaces from sparse, usually binary, and noisy input data. The method consists of three elements:

- The encoding of local image information into a tensor field.
- A model of the continuation of an elongated structure in an image
- A mechanism to communicate between different features in the image.

The name "tensor voting" refers to the fact that the information is encoded in tensors that communicate by means of a *voting* process. Tensors receive more votes if they are more likely to belong to an important perceptual structure in the image. Tensors with the most votes are the "winners". It can more or less be compared with political elections, where politicians with most votes win a seat in the parliament.

Various applications of the TV technique can be found, including (all described in [32]) extraction of object shapes out of noisy 3D data, shape from stereo, and motion estimation. Furthermore, TV was investigated at Philips Research, Paris, for the detection of guide wires [28]. Conclusion of that work is that the technique is promising for noisy guide wire images. However, only the generation of feature images using TV was investigated. The added value of the current work is that TV and related techniques are studied in more detail, a steerable TV scheme is derived, it is used in the application of EP catheters, and that the result of TV is also used in the extraction step (Chapter 5).

In the forthcoming subsections the different elements of the method, as mentioned above, are explained. Afterwards, an example is given, and the weak and strong points will be discussed. Note that in this work, we do not literally repeat the way TV is described by Medioni et al. For reasons of clarity, and for application-specific reasons, we describe some aspects in a different way and omit some aspects. For instance, we will only describe two-dimensional TV and omit the three-dimensional case.

4.2.1 Local image feature description

In tensor voting, the input data is encoded into a field of second order symmetric and real-valued tensors (on a discrete grid, equivalent to the pixel grid of an image, and defined on the bounded image domain Ω). A tensor in this field has the following form

$$\mathrm{T} = \begin{pmatrix} t_{11} & t_{12} \\ t_{12} & t_{22} \end{pmatrix} = \begin{pmatrix} \mathbf{e}_1 & \mathbf{e}_2 \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} \mathbf{e}_1^T \\ \mathbf{e}_2^T \end{pmatrix}, \tag{4.1}$$

where λ_1 and λ_2 are the eigenvalues ($\lambda_1 \geq \lambda_2$), and $\mathbf{e}_1$ and $\mathbf{e}_2$ are the orthonormal eigenvectors ($\mathbf{e}_1 \perp \mathbf{e}_2$, $||\mathbf{e}_1|| = 1$, and $||\mathbf{e}_2|| = 1$). All tensors must be semi-positive definite, i.e. $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$, because of the way the eigenvalues are interpreted. Such a tensor of rank 2 is a 2×2 matrix. In the following, when we talk about "tensor", we will always refer to this simple type of tensor.

We will first give some mathematical details about this representation. Then the interpretation that Medioni et al. assign to the eigenvalues and eigenvectors will be explained.

Rotational characteristics

To understand why this tensor representation is chosen, we will first have a look at the rotational characteristics of the tensor. A tensor A can be rotated by an angle θ as follows

$$\mathrm{A}_\theta = \mathrm{R}_\theta \mathrm{A}\,\mathrm{R}_\theta^{\mathrm{T}} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}, \tag{4.2}$$

where R_θ is the rotation matrix. It is more convenient to rewrite a tensor as a "vector"

$$\mathrm{T}' = \begin{pmatrix} t_{11} \\ t_{12} \\ t_{22} \end{pmatrix}, \tag{4.3}$$

where the accent symbol indicates the use of vector notation instead of the "normal" tensor notation. Now we can write tensor rotation as a regular matrix multiplication

$$\mathrm{A}'_\theta = \mathrm{R}'_\theta \mathrm{A}' = \begin{pmatrix} \frac{1}{2}(1+\cos(2\theta)) & -\sin(2\theta) & \frac{1}{2}(1-\cos(2\theta)) \\ \frac{1}{2}\sin(2\theta) & \cos(2\theta) & -\frac{1}{2}\sin(2\theta) \\ \frac{1}{2}(1-\cos(2\theta)) & \sin(2\theta) & \frac{1}{2}(1+\cos(2\theta)) \end{pmatrix} \begin{pmatrix} a_{11} \\ a_{12} \\ a_{22} \end{pmatrix}. \tag{4.4}$$

R'_θ is a *reducible representation* of SO(2) (see Subsection 3.5.1). We can make the *irreducible* constituents visible by calculating the eigenvalues and eigenvectors of R'_θ, leading to the following similarity transformation

$$\mathrm{A}'_\theta = \underbrace{\begin{pmatrix} 1 & 1 & 1 \\ 0 & i & -i \\ 1 & -1 & -1 \end{pmatrix}}_{\mathrm{S}} \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & e^{-2i\theta} & 0 \\ 0 & 0 & e^{2i\theta} \end{pmatrix}}_{\mathrm{R}'_{\mathrm{ir},\theta}} \underbrace{\begin{pmatrix} \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{4} & -\frac{i}{2} & -\frac{1}{4} \\ \frac{1}{4} & \frac{i}{2} & -\frac{1}{4} \end{pmatrix}}_{\mathrm{S}^{-1}} \begin{pmatrix} a_{11} \\ a_{12} \\ a_{22} \end{pmatrix}, \tag{4.5}$$

where the columns in S are the tensors that form invariant subspaces under rotation. The rotation matrix $\mathrm{R}'_{\mathrm{ir},\theta}$ in this basis is made up by a direct sum of one-dimensional irreducible representations. It can be seen that the tensor A contains an $m = 0$, $m = -2$, and $m = 2$ component (i.e., $\mathrm{R}'_{\mathrm{ir},\theta}$ contains $e^{im\theta}$ for $m = 0, -2, 2$). More intuitively it can be said that the tensor contains an isotropic part, and an 180°-symmetric part. In normal tensor notation, the previous equation can also be written as

$$\begin{aligned} \mathrm{A}_\theta = {} & \underbrace{\frac{1}{2}(a_{11}+a_{22})}_{A_0} \underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}}_{\mathrm{H}_0} + e^{-2i\theta} \underbrace{\frac{1}{4}(a_{11}-a_{22}-2i\,a_{12})}_{A_{-2}} \underbrace{\begin{pmatrix} 1 & i \\ i & -1 \end{pmatrix}}_{\mathrm{H}_{-2}} \\ & + e^{2i\theta} \underbrace{\frac{1}{4}(a_{11}-a_{22}+2i\,a_{12})}_{A_2} \underbrace{\begin{pmatrix} 1 & -i \\ -i & -1 \end{pmatrix}}_{\mathrm{H}_2}. \end{aligned} \tag{4.6}$$

where A_0, A_2, and A_{-2} are the different m-components of the tensor, viz. $\begin{pmatrix} A_0 & A_{-2} & A_2 \end{pmatrix}^{\mathrm{T}} = S^{-1}A'$, and H_0, H_2, and H_{-2} form the basis for the irreducible m-components. Later on in this chapter, we will see the benefit of this irreducible representation.

Note that if all components of A are real-valued, the previous equation can also be written as

$$\mathrm{A}_\theta = A_0\,\mathrm{H}_0 + \mathrm{Re}\left(e^{2i\theta} A_{|2|}\,\mathrm{H}_2\right), \tag{4.7}$$

where $A_{|2|} = 2A_2 = \frac{1}{2}(a_{11} - a_{22} + 2i\,a_{12})$. Because the $m = 2$ and $m = -2$ component are now combined, we will denote this as the $|m| = 2$ component.

There is a clear relationship between the different m-components and the eigenvalues and eigenvectors of the tensor. The eigenvalues are given by

$$\begin{aligned}\lambda_1 &= \frac{1}{2}\left(a_{11} + a_{22} + \sqrt{4\,a_{12}^2 + (a_{11} - a_{22})^2}\right) = A_0 + |A_{|2|}|,\\ \lambda_2 &= \frac{1}{2}\left(a_{11} + a_{22} - \sqrt{4\,a_{12}^2 + (a_{11} - a_{22})^2}\right) = A_0 - |A_{|2|}|,\end{aligned} \tag{4.8}$$

so the sum of the eigenvalues $\frac{1}{2}(\lambda_1 + \lambda_2) = \frac{1}{2}(a_{11} + a_{22}) = A_0$ corresponds to the $m = 0$ part of the tensor. The difference of the eigenvalues $\frac{1}{2}(\lambda_1 - \lambda_2) = \frac{1}{2}\sqrt{4\,a_{12}^2 + (a_{11} - a_{22})^2} = |A_{|2|}|$ corresponds to the *absolute value* of the $|m| = 2$ part of the tensor[1]. The angle of the first eigenvector is given by

$$\begin{aligned}\angle\mathbf{e}_1 &= \arg\left(a_{11} - a_{22} + \sqrt{4\,a_{12}^2 + (a_{11} - a_{22})^2} + 2i\,a_{12}\right)\\ &= \frac{1}{2}\arg\left(a_{11} - a_{22} + 2i\,a_{12}\right) = \frac{1}{2}\arg(A_{|2|}),\end{aligned} \tag{4.9}$$

from which it becomes apparent that the angle of the eigenvector is equivalent to the *argument* of $|m| = 2$ part of the tensor divided by 2. The angle of the second eigenvector will thus be $\angle\mathbf{e}_2 = \frac{\pi}{2} + \frac{1}{2}\arg(A_{|2|})$.

Interpretation of information contained by a tensor

Medioni et al. assign the following interpretation to the information that is contained in a tensor:

- $\lambda_1 - \lambda_2$ (or the $|m| = 2$ part) is the anisotropy, and is interpreted as a measure for the *orientation certainty*, in the direction $\angle\mathbf{e}_1$. Medioni et al. call it the *curveness* or *stickness.*
- λ_2 is interpreted as a measure for the *orientation uncertainty.* Medioni et al. call this the *pointness* or *ballness.*

A tensor can be illustrated graphically with an ellipse, see Figure 4.6. There are two extreme cases: the *stick tensor* ($\lambda_2 = 0$, $\lambda_1 > 0$, or alternatively $A_0 = |A_{|2|}|$) and the *ball tensor* ($\lambda_1 = \lambda_2$, or alternatively $A_{|2|} = 0$). All other tensors can be decomposed in those two elements, as follows

$$\mathrm{A} = \mathrm{A_S} + \mathrm{A_B} = (\lambda_1 - \lambda_2)\mathbf{e}_1\mathbf{e}_1^T + \lambda_2\left(\mathbf{e}_1\mathbf{e}_1^T + \mathbf{e}_2\mathbf{e}_2^T\right). \tag{4.10}$$

The TV technique takes a tensor field (say T) as input, and generates a similar tensor field (U) as output. The output field is a context enhanced version of the input field, achieved by communication between spatially neighboring tensors, as will be described in detail in the next sections.

The way the input tensor field should be encoded is left unaddressed in the framework, because it is application-specific. Usually, some kind of threshold is used to get a *sparse* set of data, which means that most tensors in the tensor field are zero.

Tensor addition

To understand tensor voting it is important to realize what happens when adding up tensors. Suppose we have two tensors A and B. The sum S = A + B can be written as (using the decomposition of Equation 4.7)

$$S = (A_0 + B_0)\,\mathrm{H}_0 + \mathrm{Re}\left((A_{|2|} + B_{|2|})\mathrm{H}_2\right). \tag{4.11}$$

Some special cases to clarify tensor addition (see Figure 4.7):

---

[1] In terms of trace and determinant of the tensor A we have the following relationships: $A_0 = \frac{1}{2}\mathrm{Tr}\,\mathrm{A}$, and $A_2\,A_{-2} = \frac{1}{16}((\mathrm{Tr}\,\mathrm{A})^2 - 4\det\mathrm{A})$.

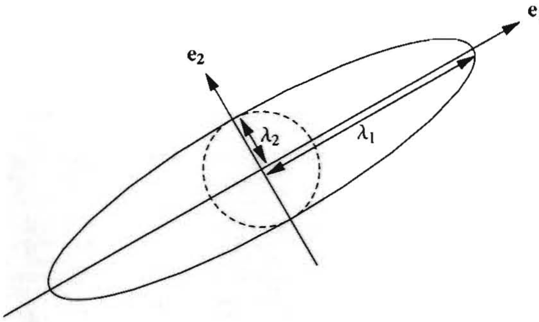


Figure 4.6: *Graphical representation of a second order symmetric semi-positive definite tensor.*

- Addition of two ball tensors. In this case $A_{|2|} = B_{|2|} = 0$, so only $A_0 + B_0$ have influence, resulting in a ball tensor with *higher* saliency.
- Addition of two stick tensors with equal orientation. For a stick tensor, $A_0 = |A_{|2|}|$. $A_{|2|}$ and $B_{|2|}$ have the same complex phase for equally directed tensors, so for the result $S_0 = |S_{|2|}|$ will still hold. So the result is a stick tensor again, but with *summed* stick saliency.
- Addition of two stick tensors that are orthogonal. In this case, $A_{|2|} = -B_{|2|}$, meaning that the $|m| = 2$ component of the resulting tensor will be zero, and only the $m = 0$ component will remain. So the result of this addition will be a ball tensor.

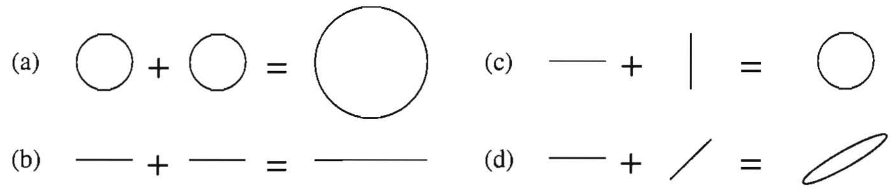


Figure 4.7: *Four examples of tensor addition. (a) Addition of two ball tensors results in a ball tensor. (b) Addition of two stick tensors with equal direction results in a stick tensor. (c) Addition of two orthogonal stick tensors results in a ball tensor. (d) Addition of two non-orthogonal stick tensors, leading to a tensor with a ball and stick component.*

4.2.2 Voting fields

In the 2D case Medioni et al. use two different *voting fields*, which specify how local features communicate: the *stick field* and the *ball field*. Those fields in principle have the same representation as the input data, i.e. they are tensor fields as well. The voting fields determine the way the local image structure information is communicated, using the communication scheme described in the next subsection.

The stick field

The stick field renders a model for the possible continuations of an oriented structure. Suppose we have an oriented feature at location $\mathbf{v}_o$ with orientation 0 (i.e. horizontal orientation), see Figure 4.8. We want to know:

- What is the most likely orientation of a feature at position $\mathbf{c}$ relative from $\mathbf{v}_o$, if the features would belong to the same object?

- What is the probability that the feature at $\mathbf{v}_o$ and at position $\mathbf{c}$ belong to the same object in the image?

Below, we will first derive the most likely orientation V_{cc} and then the probability distribution S.

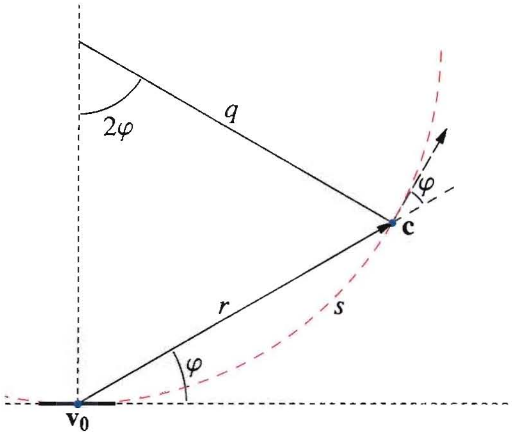


Figure 4.8: *Geometric reasoning for deriving the stick weighting function. See text for details.*

For the most likely orientation V_{cc}, we use the Gestalt principle of good continuation, which implies that line segments should be as smooth as possible. The smoothest line segment, i.e. the line segment with smallest total curvature, between $\mathbf{v}_o$ with an imposed orientation and $\mathbf{c}$ is a circular arc. Therefore, it is assumed in the model that the connecting path in between follows a circular arc. This constraint is called *cocircularity* [17, Chapter 4], see Figure 4.9a. In Figure 4.8 it can be seen that in case of cocircularity the orientation at position $\mathbf{c}$ must be $2\angle\mathbf{c} = 2\varphi$, yielding the vector $\binom{\cos 2\varphi}{\sin 2\varphi}$. To encode the constraint in a tensor field consisting entirely of stick tensors, we take the outer product of this vector with itself. This yields the following tensor field, expressed in spatial polar coordinates and cartesian tangent space

$$\tilde{\mathrm{V}}_{\mathrm{cc}}(r,\varphi) = \begin{pmatrix}\cos 2\varphi\\ \sin 2\varphi\end{pmatrix} \cdot \begin{pmatrix}\cos 2\varphi\\ \sin 2\varphi\end{pmatrix}^{\mathrm{T}} = \begin{pmatrix}\frac{1}{2}\left(1+\cos\left(4\varphi\right)\right) & \frac{1}{2}\sin\left(4\varphi\right)\\ \frac{1}{2}\sin\left(4\varphi\right) & \frac{1}{2}\left(1-\cos\left(4\varphi\right)\right)\end{pmatrix}. \tag{4.12}$$

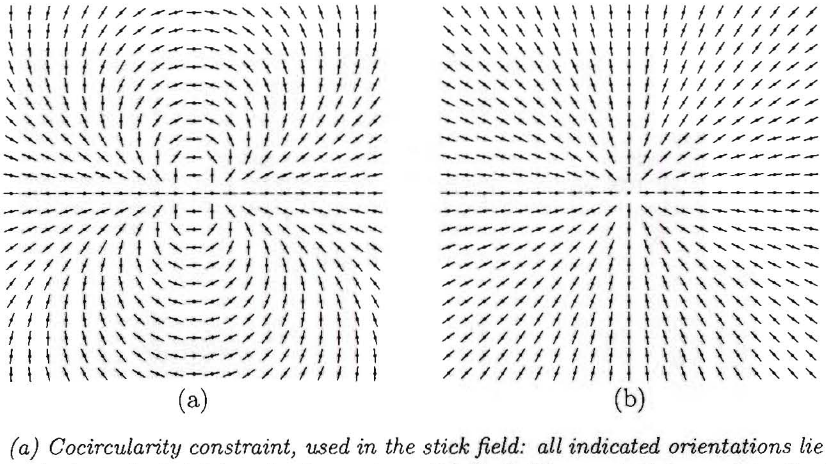

(a) (b)

Figure 4.9: *(a) Cocircularity constraint, used in the stick field: all indicated orientations lie on a circular arc together with the oriented token in the center. (b) Radiality constraint, used in the ball field: all indicated orientations lie on a straight line together with the non-oriented token in the center.*

For the derivation of the probability distributions, again the Gestalt principles are used: proximity implies that there should be a decay as function of the length of the connecting circular arc, and good continuation suggests there should be a decay as function of the curvature of the connecting circular arc. The idea is illustrated in Figure 4.8. The radius q of the circular arc connecting $\mathbf{v}_o$ and $\mathbf{c}$ is: $q = \frac{r}{2\sin\varphi}$, where $r = |\mathbf{c}|$ and $\varphi = \angle\mathbf{c}$. So the length of the circular arc is

$$s = 2\varphi q = \frac{\varphi r}{\sin\varphi}, \tag{4.13}$$

and the curvature is

$$\kappa = \frac{1}{q} = \frac{2\sin\varphi}{r}. \tag{4.14}$$

A Gaussian decay with respect to these two parameters is assumed, which results in the following decay function

$$S(s,\kappa) = e^{-\frac{s^2}{\sigma_{\text{ctx}}^2} - p\kappa^2\sigma_{\text{ctx}}^2}, \tag{4.15}$$

where σ_{ctx} is the scale of the voting field, and p is a dimensionless constant describing the relative weight of the curvature. In polar coordinates this yields

$$\tilde{S}(r,\varphi) = e^{-\left(\frac{\varphi r}{\sigma_{\text{ctx}}\sin\varphi}\right)^2 - p\left(\frac{2\sigma_{\text{ctx}}\sin\varphi}{r}\right)^2}. \tag{4.16}$$

Medioni et al. state that points above an below the main diagonals in the field are too unlikely to belong to the same structure as the point in the center of the field. Therefore the field is truncated for these values of φ, by multiplying S by truncation function Υ

$$\Upsilon(\varphi) = \begin{cases} 1 & \text{if } -\pi/4 \leq \varphi \leq \pi/4 \text{ and } \varphi \geq 3\pi/4 \text{ and } \varphi \leq -3\pi/4 \\ 0 & \text{otherwise,} \end{cases} \tag{4.17}$$

with $-\pi < \varphi \leq \pi$.

Combining the directional part $\tilde{\mathrm{V}}_{\text{cc}}$, the weighting function $\tilde{S}$, and truncation function Υ yields the following voting field

$$\tilde{\mathrm{V}}_{\text{S}}'(r,\varphi) = \tilde{\mathrm{V}}_{\text{cc}}'(r,\varphi)\,\tilde{S}(r,\varphi)\,\Upsilon(\varphi) = \begin{pmatrix} \frac{1}{2}\,(1+\cos(4\varphi)) \\ \frac{1}{2}\sin(4\varphi) \\ \frac{1}{2}\,(1-\cos(4\varphi)) \end{pmatrix} e^{-\left(\frac{\varphi r}{\sigma_{\text{ctx}}\sin\varphi}\right)^2 - p\left(\frac{2\sigma_{\text{ctx}}\sin\varphi}{r}\right)^2}\,\Upsilon(\varphi). \tag{4.18}$$

In Figure 4.10a, the resulting stick field is shown.

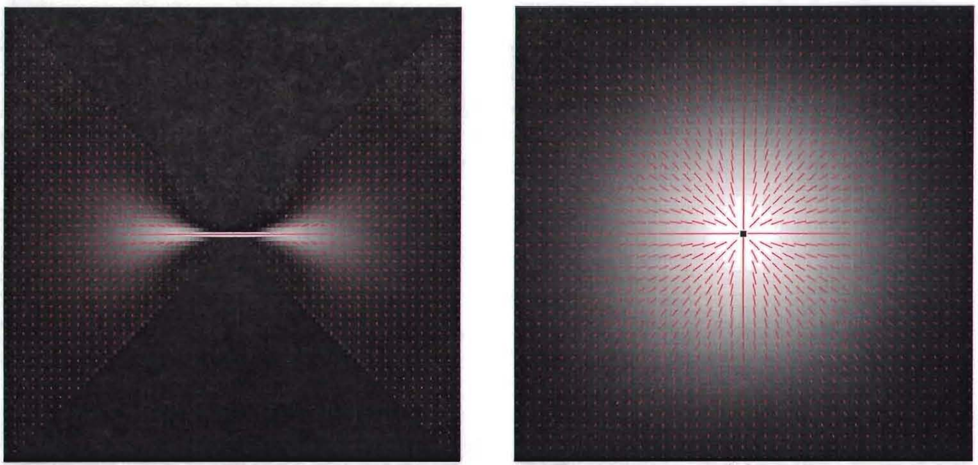

(a) (b)

Figure 4.10: *Voting fields as used in tensor voting. (a) The stick voting field. (b) The ball voting field.*

The stick voting field needs to be rotated. We need to rotate the field itself, *and* the tensors in the field. A rotation by θ is calculated as follows

$$\tilde{\mathrm{V}}'_{\mathrm{S},\theta}(r,\varphi) = \mathrm{R}'_\theta \tilde{\mathrm{V}}'_{\mathrm{S}}(r,\varphi-\theta), \tag{4.19}$$

where R'_θ is the rotation matrix as defined in Equation 4.4.

The ball field

Medioni et al. also introduce an isotropic voting field: the *ball field*. The idea of the ball field is to use it if the input tensor field does not supply orientational information, i.e. if the input is formed by non-oriented tensor tokens. In that case the stick voting field is not applicable, because we do not know how to align the field if no orientational information is available.

When no orientation information is available beforehand, a sensible prior is to use the *radiality* constraint, which is shown in Figure 4.9b. Given an unknown orientation in the center $\mathbf{v}_0$, the most likely orientation at a position $\mathbf{c}$ relative from the center, is specified by $\mathbf{c}/|\mathbf{c}|$. The underlying assumption is that the most likely connection between $\mathbf{v}_0$ and $\mathbf{c}$ is formed by a straight line. This results in the following tensor field

$$\tilde{\mathrm{V}}_{\mathrm{rad}}(r,\varphi) = \begin{pmatrix}\cos\varphi\\ \sin\varphi\end{pmatrix} \cdot \begin{pmatrix}\cos\varphi\\ \sin\varphi\end{pmatrix}^{\mathrm{T}} = \begin{pmatrix}\frac{1}{2}\left(1+\cos\left(2\varphi\right)\right) & \frac{1}{2}\sin\left(2\varphi\right)\\ \frac{1}{2}\sin\left(2\varphi\right) & \frac{1}{2}\left(1-\cos\left(2\varphi\right)\right)\end{pmatrix}. \tag{4.20}$$

As weighting function, a Gaussian decay as function of the radius is the most natural choice. Medioni et all derive this by integrating the weighting function of the stick field (Equation 4.16) over φ. For the ball field we yield the following expression (in polar coordinates)

$$\tilde{\mathrm{V}}'_{\mathrm{B}}(r,\varphi) = \begin{pmatrix}\frac{1}{2}\left(1+\cos\left(2\varphi\right)\right)\\ \frac{1}{2}\sin\left(2\varphi\right)\\ \frac{1}{2}\left(1-\cos\left(2\varphi\right)\right)\end{pmatrix} e^{-\frac{r^2}{2\sigma_{\mathrm{ctx}}^2}}. \tag{4.21}$$

The resulting field is shown in Figure 4.10b.

4.2.3 The voting mechanism

We introduced a representation of local image structure, and derived voting fields based on Gestalt principles. This section describes how the context communication between the different tensors is achieved in the cases of stick and ball voting.

Stick voting

Stick voting is illustrated in Figure 4.11 for a simple example with only a sparse number of tensor tokens. For every (nonzero) tensor (e.g. A) in the tensor field, the stick field is centered at the position of the tensor and rotated to be aligned with the first eigenvector $\mathbf{e}_1$ of the tensor. All tensors in a certain neighborhood (determined by the context scale σ_{ctx}) of this tensor receive a *vote contribution*, by addition of the value of the voting field weighted by the anisotropy part of the tensor, $\lambda_1 - \lambda_2$ of A. In other words: the information contained by the tensor A is *broadcasted* to the surrounding. In Figure 4.11 this broadcasting process is illustrated for 3 tensors, but this is done for every nonzero tensor, and the resulting vote contributions at every position are added up.

The voting process as described above, leads to the following operational definition for dense stick voting[2]

$$\mathrm{U}(x,y) = \iint_\Omega \left(\lambda_1(x',y') - \lambda_2(x',y')\right)\ \mathrm{V}_{\angle\mathbf{e}_1(x',y')}(x'-x,y'-y)\mathrm{d}x'\mathrm{d}y', \tag{4.22}$$

where $\mathrm{V}_{\angle\mathbf{e}_1(x',y')}$ is the voting field, expressed in cartesian coordinates and rotated over $\angle\mathbf{e}_1(x',y')$. $\lambda_1(x,y)$, $\lambda_2(x,y)$, and $\mathbf{e}_1(x,y)$ are the eigenvalues resp eigenvectors of the tensors in input tensor field T.

---

[2] Surprisingly, nowhere in the TV literature the operational definition is really defined. It is normally only described as pseudo-code.

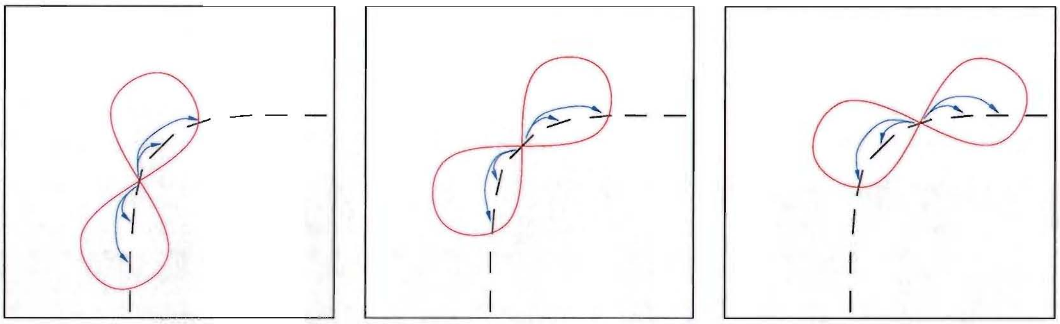


Figure 4.11: *Illustration of context communication within TV: the oriented tensors communicate with each other using the voting field, and in this way strengthen each other. For clarity, this figures illustrate a* sparse *stick tensor voting, i.e. there are only nonzero tensors at a limited number of positions.*

Ball voting

Because of the isotropy, the operational definition for ball voting is simpler. In this case the contribution of a tensor to its neighbors is weighted by $\lambda_2(x, y)$

$$\mathrm{U}(x,y) = \iint_{\Omega} \lambda_2(x',y')\, \mathrm{V_B}(x'-x, y'-y)\mathrm{d}x'\mathrm{d}y', \tag{4.23}$$

where $\mathrm{V_B}(x, y)$ is the ball voting field, in cartesian coordinates. $\lambda_2(x, y)$ are the second eigenvalues of the tensors in input tensor field T.

Context broadcast versus collect

A note should be made on the *broadcasting* process. As shown in Figure 4.11, the stick field is aligned with the principal direction of a tensor and then all neighboring tensors receive a contribution of this field. This is the way Medioni et al. do it, but the other way around is also possible: we could align the field with the principal direction of a tensor and then *collect* neighboring tensors, by rotating and attenuating them according to the voting field.

These two mechanisms are different in the case of an anisotropic voting field. This is illustrated in Figure 4.12: in the case of a tensor token that has an invalid orientation, the collection mechanism would not be able to correct this, because the voting field will be aligned with the wrong orientation. In the case of broadcasting, however, the tensor token will receive a lot of votes from neighboring tensors that will very likely belong to the same structure, so the orientation of the tensor token will be corrected. Of course, the tensor with invalid orientation will cause a broadcast in the wrong direction, causing artefacts, but as long as there are a limited number of erroneous tensors, this broadcast will usually not be very strong.

In conclusion, for now the *broadcast* mechanism is most appropriate to use. Other more advanced communication mechanisms could be considered, for instance a combination of broadcast and collection. This is interesting for further research.

Sparse versus dense voting

The TV method distinguishes two types of voting:

- Sparse voting: only positions with nonzero tensors are updated, so the tensor field remains sparse. The function of this type of voting is to *enhance* orientational information of the sparse set of tensors.
- Dense voting: all positions in the grid are updated resulting in a dense tensor field. The function of this type of voting is to get orientational information everywhere, to be able to *extract* global features.

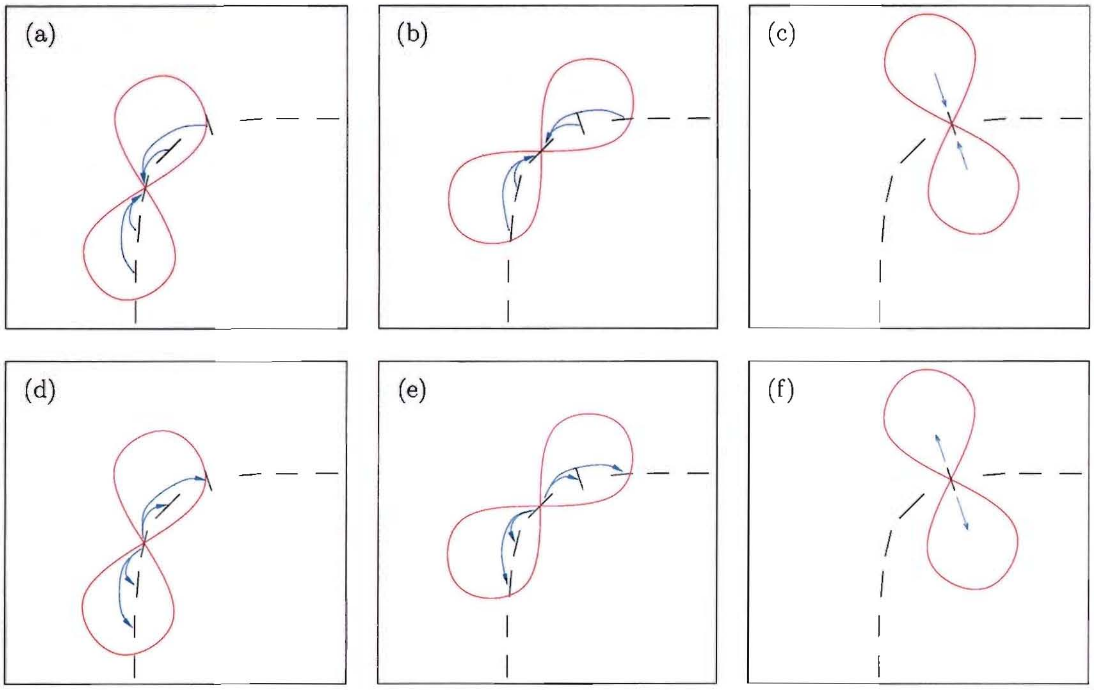


Figure 4.12: *Illustration of the difference between context collection (a-c) and broadcast (d-e). One tensor token has an inconsistent orientation. In the case of broadcast, the tensors broadcasted in (d) and (e) will correct the orientation. In the case of collection, the tensors collected in (c) will never be capable of correcting the orientation.*

Processing of the vote result

After the voting process is performed, the following information is extracted from the resulting tensor field $U(x, y)$ for further processing:

- The *ball* or *junction saliency map*, corresponding to λ_2 of the tensors in U.
- The *stick* or *curve map*, corresponding to $\lambda_1 - \lambda_2$ of the tensors in U.
- The *stick orientation map*, corresponding to $\angle \mathbf{e}_1$ of the tensors in U.

From these maps, the desired features are extracted by applying non-maximum suppression (with the use of the orientation map in case of the stick map) and possibly thresholding. Some more details are given in [32].

4.2.4 Example on an artificial image

Figure 4.13 shows an example to illustrate the TV process. In this case the input is a sparse set of non-oriented tensor tokens (indicated by white pixels in Figure 4.13a). To assign orientation to these tensors, first a sparse ball voting is performed (Figure 4.13c). Then, a dense stick voting is performed and the resulting tensors are decomposed, which results in stick and ball saliency maps (Figure 4.13d-e). From these maps, the local maxima can be extracted to find the desired curves and junctions (Figure 4.13f).

In Figure 4.14, TV is applied to the same artificial image, but now with added uncorrelated noise. Medioni et al. add noise by just adding nonzero tokens at random positions in the image. In the example of Figure 4.14, 7 noise points are added for every correct point, so 1 out of 8 nonzero tensors is correct. It can be seen that the lines, and 4 out of 5 of the junctions, are still extracted quite well.

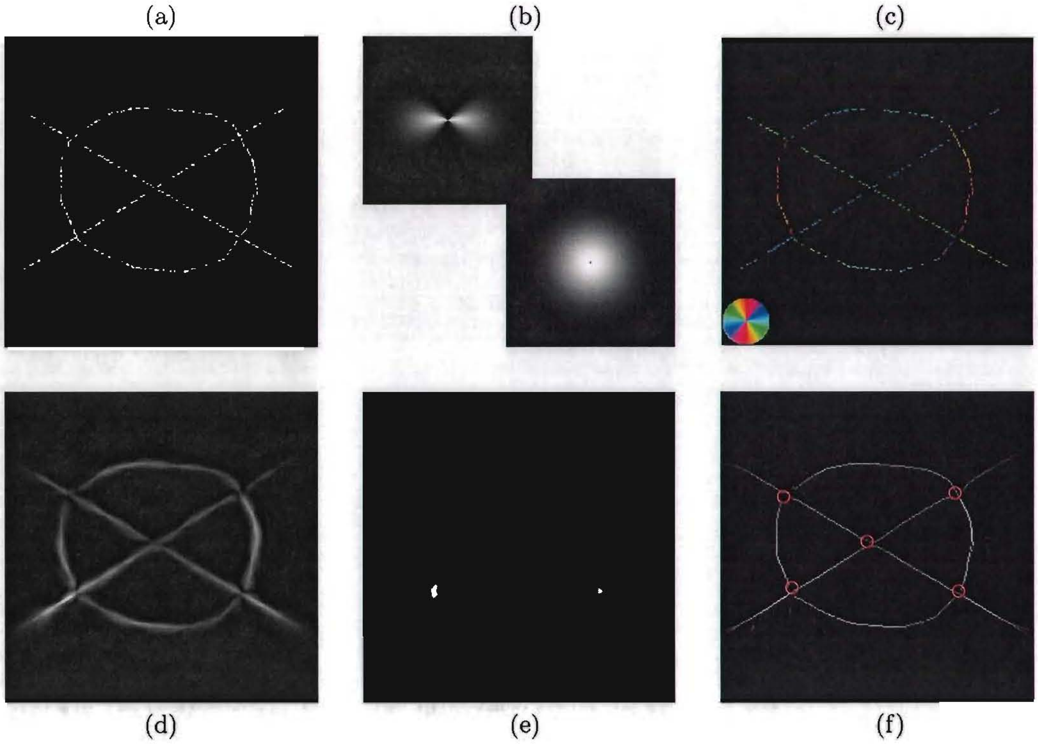


Figure 4.13: *Example of TV on an artificial image. (a) Source, consisting of a sparse set of non-oriented tokens. (b) Used stick voting and ball voting fields, displayed on the same size relative to the image. (c) Result after sparse ball voting, where the color indicates the inferred orientation coded as indicated by the disc at the bottom left. (d) Stick saliency ($\lambda_1 - \lambda_2$) result after dense stick voting. (e) Ball saliency (λ_2) result after dense stick voting. (f) Extraction of local maxima of the stick map, resulting in the correct line segments. The 5 superimposed circles indicate the 5 highest local maxima extracted from the ball map.*

4.2.5 Strong and weak points

Advantages of TV are:

1. The method makes the grouping of local structures to more global structures easier: for instance, gaps are filled and false positives are attenuated. It is a powerful tool for perceptual grouping.
2. There are more related techniques described in literature (e.g. [18, 17]) that also use "8-shaped voting fields" for perceptual grouping. The major difference of TV is the use of rank 2 tensors, which encode orientation certainty *and* uncertainty.
3. The basic TV scheme is in principle non-iterative, which means that the grouping can be performed in one (or a few) steps. This means that the method is efficient, relative to iterative context enhancing methods (e.g. based on Markov random fields [15]).

However, the approach of Medioni et al. also has weak points:

1. As input, normally a sparse set of tensor tokens is used. Often these tensor tokens are even binary. The way to transform an image to this sparse set of tokens is said to be application-specific. In this way the problem of inferring the desired structure from a noisy input image is in fact shifted to this preprocessing step. For example, in our application it is hard to first make the data sparse in a reliable way. We observed that in noisy EP catheter images, the response of the local filter on the catheter is not clear enough to robustly select a sparse set of reliable tokens. A good threshold value, for instance, is hard to choose, because it differs a lot from image to image and even for different image regions.

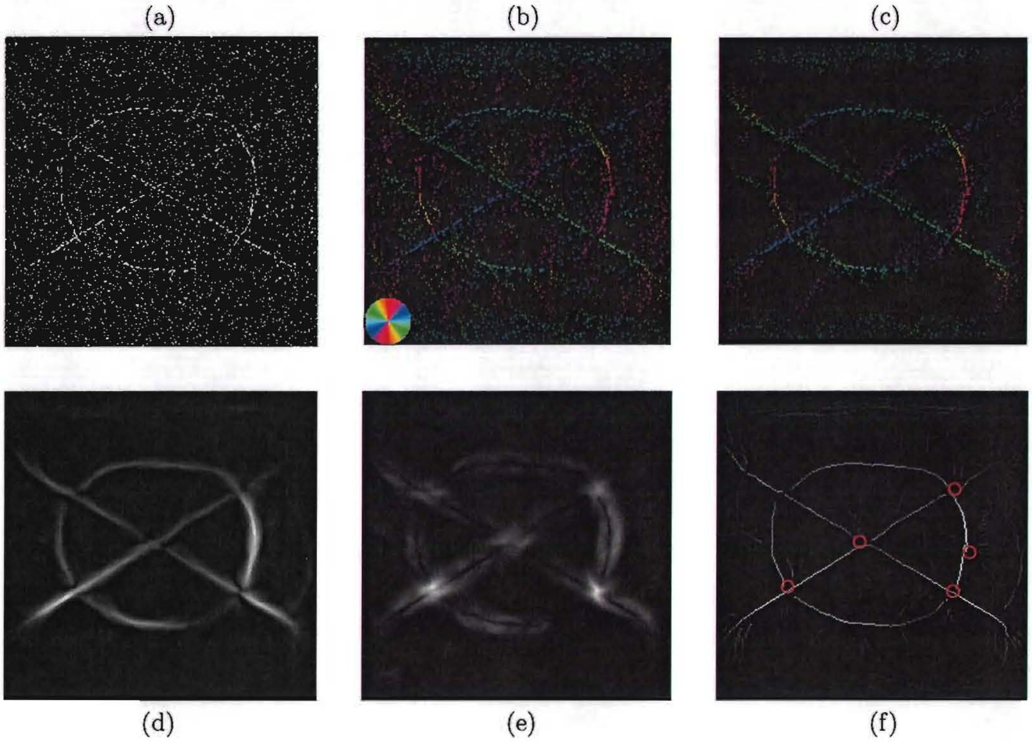


Figure 4.14: *Example of TV on an artificial image, with added noise. (a) Noisy input, consisting of a sparse set of non-oriented tokens. (b) Result after sparse ball voting, where the color indicates the inferred orientation, coded as indicated by the disc at the bottom left. (c) Result after an additional sparse stick voting, which leads to a reduction of erroneous tensor tokens. (d) Stick saliency ($\lambda_1 - \lambda_2$) result after dense stick voting. (e) Ball saliency (λ_2) result after dense stick voting. (f) Extraction of local maxima of the stick map, and the 5 highest local maxima of the ball map (indicated by red circles).*

2. The stick voting field has some undesired characteristics. It is *not steerable* so it needs to be explicitly rotated. It is very thin in the center, leading to aliasing artefacts when rotating it on a pixel grid[3].

3. *End-points of lines* in the image are not well-preserved, because this is not modelled in the voting field and because rank 2 tensors are used.

4. The method does not work well for structures with very *high curvature*, because these structures are "rejected" by the voting field. In our application this can be a problem because of the projection from 3D to 2D (catheters with a kink in the image).

5. According to Medioni et al., the ball saliency (λ_2) of the resulting tensor field is a good measure for *junction detection*. Junctions are extracted by applying a threshold on the ball saliency. We think, however, that this is not a reliable measure for detecting junctions. It only works well for junctions where the incoming lines are nearly orthogonal to each other. In the case of catheters this is often not the case. Also, the ball saliency can give a high response in noisy parts of the image without oriented structures, where votes are received from all different directions.

6. Medioni et al. claim that they have only one free parameter: scale. This is simply not true, because also the width of the stick voting field (p in Equation 4.18) is an important parameter. The preprocessing step and the structure extraction step will also require some parameters.

---

[3] It is not clear how Medioni et al. handle this.

4.2.6 The use of tensor voting for catheter detection

Although quite some weak points were indicated in the previous subsection, TV is an interesting technique to use for our catheter detection problem, for the following reasons:

- The expectation is that we will be able to extract catheters under lower dose conditions as compared to only using local feature detectors. We expect that the TV step will amplify the catheters, since catheters are long and smooth elongated structures in the image. False responses that are caused by the local ridge detector will be attenuated. This is especially important because we want to detect the catheters without an initial seed position, implying that we have to search through a fairly large search space. It will be experimentally verified in Chapter 6 whether this expectation is true.
- Although quite expensive, the method is still relatively cheap compared to other perceptual grouping methods, for instance based on orientation bundles [22] and iterative schemes.

In the context of EP catheter detection, we expect to solve or handle the weak points enumerated in Subsection 4.2.5 in the following way:

1. Instead of a sparse, binary, input tensor map we will use the dense feature map of Chapter 3 directly as input to TV. The input tensors $\mathrm{T}(x, y)$ will all be stick tensors, such that

$$\lambda_1 - \lambda_2 = r \quad \text{and} \quad \angle \mathbf{e}_1 = \psi, \tag{4.24}$$

   where r the local ridgeness value and ψ the local orientation from Chapter 3. The second eigenvalue λ_2 will be set to zero, because we will not use ball voting.

2. Instead of the standard stick TV field, we will use a *steerable* voting field (Section 4.3) and a scheme to efficiently use this steerable voting field. This has computational advantages, especially in the case of dense input data.

3. Concerning the bad preservation of end-points: we will not try to solve this problem in general, but instead use the application-specific knowledge that catheters in an X-ray image always have an electrode or ablation tip at one end-point, and that the other end-point is situated at one of the borders of the image.

4. Concerning catheters with a kink because of the 3D to 2D projection: it is difficult to take this into account in the voting field. We would have to let go the assumption of small curvature, while this property will hold almost everywhere else. A kink will seldomly occur on more than one place on the catheter simultaneously, so we think it can be handled by the high-level extraction stage. Also, the use of temporal tracking will make the method more robust to kinks in catheters.

5. Concerning the detection of junctions: in our application the proper localization of junctions is not the main issue. We have to detect the catheters and not the junctions.

6. Concerning the problem of scale: we will not try to solve this problem. We do not expect a considerable performance gain for this specific application. It would just lead to a considerably higher computational work load.

4.3 Steerable tensor voting

In this section we derive a tensor voting scheme that uses steerability. First, a steerable voting field is introduced. Then, we will show how to exploit steerability, leading to a voting scheme that we call *steerable tensor voting*.

4.3.1 Steerable voting field

The standard stick voting field of Subsection 4.2.2 is not steerable. In this section, we will derive a steerable stick voting field. For the directional part of the voting field, we again use the cocircularity constraint V_{cc}, as defined in Equation 4.12.

The derivation of the weighting part of our field is straightforward: as overall weighting of the voting field, we use the Gaussian kernel: $e^{-\frac{r^2}{2\sigma_{ctx}^2}}$. This models the gestalt principle of proximity. In order to assign heavier weight to points that can be connected with a smooth circular arc with low curvature (the good continuation principle), we multiply the overall weighting function with $\cos^{2n}\varphi$ where $n \in \mathbb{N}$ is a degree of freedom specifying the width of the field (see Figure 4.15). This degree of freedom is comparable to the p in Equation 4.16. This gives for the stick field in polar coordinates

$$\tilde{V}'(r,\varphi) = \begin{pmatrix} \frac{1}{2}\left(1+\cos\left(4\varphi\right)\right) \\ \frac{1}{2}\sin\left(4\varphi\right) \\ \frac{1}{2}\left(1-\cos\left(4\varphi\right)\right) \end{pmatrix} e^{-\frac{r^2}{2\sigma_{ctx}^2}} \cos^{2n}\varphi. \tag{4.25}$$

An example is shown in Figure 4.16.

Figure 4.15: *Weighting field of the voting field for different values of n.*

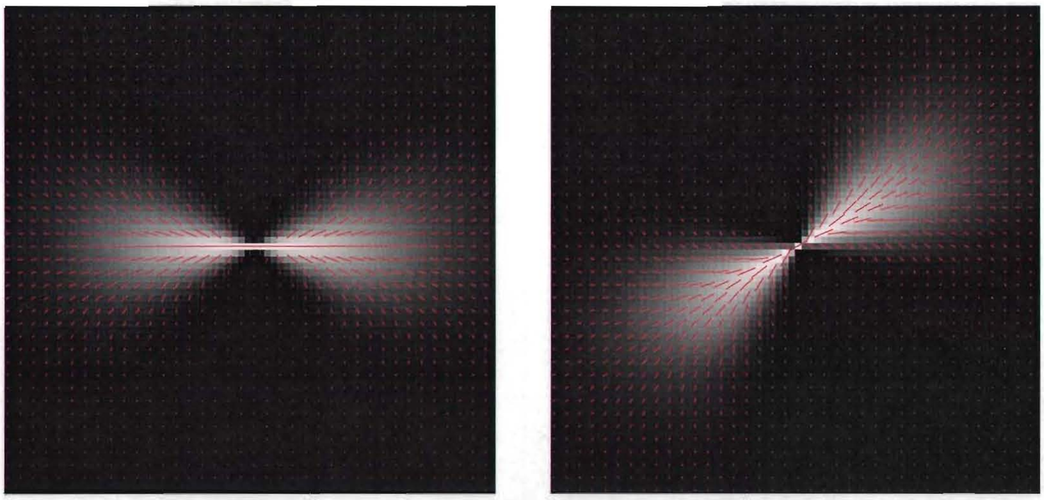


Figure 4.16: *The steerable stick voting field, in two different orientations.*

Other perceptual grouping methods use similar voting kernels, see e.g. [18, 30]. The differences with the stick field as proposed by Medioni et al. (Subsection 4.2.2) are:

- The original stick voting kernel has an exponential decay as function of the length of the circular arc s, while this field has an exponential decay as function of the radius r (Figure 4.8).

- The original kernel also has an exponential decay as function of the curvature of the circular arc, which was equal to $\frac{r}{2\sin\varphi}$ (Equation 4.14). In the current field, we do not have such an exponential decay, but instead there is a decay as function of $\cos^{2n}\varphi$.

Note, that at the moment the choice of this field is rather ad hoc. To get an optimal voting field, one could gather statistics of EP catheters in a lot of images. We did not do this, because we did not have enough images with ground truth available and because we expected only a minor improvement.

4.3.2 Derivation and implementation of steerable tensor voting

In this section we will see how we can benefit from a steerable voting field. As example, we will explain how to implement steerable TV for the field with $n = 2$ in Equation 4.25. A similar derivation can be done for any other steerable voting field (Mathematica is a useful tool to do this). The field can be written as

$$\begin{aligned}
\tilde{\mathrm{V}}'(r,\varphi) &= e^{-\frac{r^2}{2\sigma_{\mathrm{ctx}}^2}} \cos^4\varphi \begin{pmatrix} \frac{1}{2}\left(1+\cos\left(4\varphi\right)\right) \\ \frac{1}{2}\sin\left(4\varphi\right) \\ \frac{1}{2}\left(1-\cos\left(4\varphi\right)\right) \end{pmatrix} \\
&= \frac{1}{32}\, e^{-\frac{r^2}{2\sigma_{\mathrm{ctx}}^2}} \begin{pmatrix} 7+12\cos 2\varphi+8\cos 4\varphi+4\cos 6\varphi+\cos 8\varphi \\ 4\sin 2\varphi+6\sin 4\varphi+4\sin 6\varphi+\sin 8\varphi \\ 5+4\cos 2\varphi-4\cos 4\varphi-4\cos 6\varphi-\cos 8\varphi \end{pmatrix} \\
&= \frac{1}{64}\, \underbrace{e^{-\frac{r^2}{2\sigma_{\mathrm{ctx}}^2}}}_{G_\sigma(r)} \underbrace{\begin{pmatrix} 1 & 4 & 8 & 12 & 14 & 12 & 8 & 4 & 1 \\ i & 4i & 6i & 4i & 0 & -4i & -6i & -4i & -i \\ -1 & -4 & -4 & 4 & 10 & 4 & -4 & -4 & -1 \end{pmatrix}}_{\mathrm{A}} \underbrace{\begin{pmatrix} e^{-i8\varphi} \\ e^{-i6\varphi} \\ e^{-i4\varphi} \\ e^{-i2\varphi} \\ 1 \\ e^{i2\varphi} \\ e^{i4\varphi} \\ e^{i6\varphi} \\ e^{i8\varphi} \end{pmatrix}}_{\boldsymbol{\Phi}(\varphi)},
\end{aligned} \tag{4.26}$$

where $G_\sigma(r)$ is the radial function of the voting field (which is the same for all m-components in our case), $\boldsymbol{\Phi}(\varphi)$ is a vector containing all the m-components in the field, and A is a coefficient matrix containing the weights of the different m-components. Further on, we will neglect the arising factor $\frac{1}{64}$, to keep the formulas more readable. In the end, normalization factors can be added again, but omitting them only leads to a factor difference in the end result.

We have a tensor field, which means that we need to rotate the field itself *and* the tensors in the field. The rotation of the field itself is achieved by

$$\tilde{\mathrm{V}}'(r,\varphi-\theta) = \mathrm{A}\mathrm{Q}_\theta \boldsymbol{\Phi}(\varphi) G_\sigma(r), \tag{4.27}$$

with

$$\mathrm{Q}_\theta = \mathrm{diag}\left(e^{i8\theta}, e^{i6\theta}, e^{i4\theta}, e^{i2\theta}, 1, e^{-i2\theta}, e^{-i4\theta}, e^{-i6\theta}, e^{-i8\theta}\right), \tag{4.28}$$

where "diag" indicates that a square matrix is formed with the elements on the diagonal. Note, that this rotation matrix is made up by a direct sum of one-dimensional irreducible representations of SO(2).

Including the rotation of the tensors in the field and rewriting all trigonometric functions as exponentials, results in

$$\begin{aligned}\tilde{\mathrm{V}}'_\theta(r,\varphi) &= \mathrm{R}'_\theta \tilde{V}'(r,\varphi-\theta)\\ &= \mathrm{R}'_\theta \mathrm{A}\,\mathrm{Q}_\theta \mathbf{\Phi}(\varphi) G_\sigma(r)\\ &= \underbrace{\begin{pmatrix} e^{6i\theta} & ie^{6i\theta} & -e^{6i\theta}\\ 4e^{4i\theta} & 4ie^{4i\theta} & -4e^{4i\theta}\\ 6e^{2i\theta}+2e^{4i\theta} & 6ie^{2i\theta} & -6e^{2i\theta}+2e^{4i\theta}\\ 4+8e^{2i\theta} & 4i & -4+8e^{2i\theta}\\ 12+e^{-2i\theta}+e^{2i\theta} & ie^{-2i\theta}-ie^{2i\theta} & 12-e^{-2i\theta}-e^{2i\theta}\\ 4+8e^{-2i\theta} & -4i & -4+8e^{-2i\theta}\\ 6e^{-2i\theta}+2e^{-4i\theta} & -6ie^{-2i\theta} & -6e^{-2i\theta}+2e^{-4i\theta}\\ 4e^{-4i\theta} & -4ie^{-4i\theta} & -4e^{-4i\theta}\\ e^{-6i\theta} & -ie^{-6i\theta} & e^{-6i\theta} \end{pmatrix}^{\mathrm{T}}}_{\mathrm{K}(\theta)} \mathbf{\Phi}(\varphi) G_\sigma(r). \end{aligned} \tag{4.29}$$

We now separated the dependency on the rotation angle θ from the polar coordinate system (r,φ), which means the field is made steerable, where $\mathrm{K}(\theta)$ in the previous equation is the matrix of linear coefficients as function of rotation. The basis functions are given by

$$\tilde{\mathbf{W}}(r,\varphi) = \mathbf{\Phi}(\varphi) G_\sigma(r). \tag{4.30}$$

The steerable voting field can be used in the voting scheme as follows. The operational definition of stick voting is already given in Equation 4.22. We replace $\lambda_1 - \lambda_2$ by r (the input ridgeness measure) and $\angle \mathbf{e}_1$ by ψ (the estimated orientation), see Equation 4.24.

$$\mathrm{U}'(x,y) = \iint_\Omega r(x',y')\ \mathrm{V}_{\psi(x',y')}(x'-x, y'-y)\mathrm{d}x'\mathrm{d}y'. \tag{4.31}$$

Substituting the voting field with the steerable version results in

$$\begin{aligned}\mathrm{U}'(x,y) &= \iint_\Omega r(x',y')\ (\mathrm{K}(\psi(x',y'))\ \mathbf{W}(x'-x,y'-y))\\ &= \sum_{j=1}^{9} \iint_\Omega (r(x',y')\ \mathbf{k}_j(\psi(x',y')))\ W_j(x'-x,y'-y)\\ &= \sum_{j=1}^{9} \{(r\ \mathbf{k}_j(\psi)) * W_j\}(x,y)\\ &= \sum_{j=1}^{9} \begin{pmatrix} (r\,k_{j1}(\psi)) * W_j\\ (r\,k_{j2}(\psi)) * W_j\\ (r\,k_{j3}(\psi)) * W_j \end{pmatrix} (x,y), \end{aligned} \tag{4.32}$$

where $\mathbf{k}_j$ denotes the jth column in matrix K, and W_j denotes the jth component of vector $\mathbf{W}$.

Using the results of Equation 4.29 and Equation 4.32 we could directly implement a steerable TV scheme. However, $\mathrm{K}(\theta)$ is rather cumbersome. It is more convenient to calculate the m-components of the tensors. Using the similarity transformation S in Equation 4.5 (Subsection 4.2.1), we obtain for the m-components of the voting field $\tilde{\mathrm{V}}'_\theta(r,\varphi)$

$$\begin{aligned}\begin{pmatrix} \tilde{V}_0^\theta(r,\varphi)\\ \tilde{V}_{-2}^\theta(r,\varphi)\\ \tilde{V}_2^\theta(r,\varphi) \end{pmatrix} &= \mathrm{R}'_{\mathrm{ir},\theta}\mathrm{S}^{-1}\,\mathrm{A}\,\mathrm{Q}_\theta \mathbf{\Phi}(\varphi) G_\sigma(r)\\ &= \underbrace{\begin{pmatrix} 0 & 0 & 2e^{4i\theta} & 8e^{2i\theta} & 12 & 8e^{-2i\theta} & 2e^{-4i\theta} & 0 & 0\\ e^{6i\theta} & 4e^{4i\theta} & 6e^{2i\theta} & 4 & e^{-2i\theta} & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & e^{2i\theta} & 4 & 6e^{-2i\theta} & 4e^{-4i\theta} & e^{-6i\theta} \end{pmatrix}}_{\mathrm{L}(\theta)} \tilde{\mathbf{W}}(r,\varphi). \end{aligned} \tag{4.33}$$

The resulting steer coefficient matrix $\mathrm{L}(\theta)$ is much more convenient than the previous $\mathrm{K}(\theta)$: there are a lot of zero entries in the matrix and every entry only contains one $e^{im\theta}$. The operational definition of steerable TV now becomes (similar to Equation 4.32)

$$\begin{pmatrix} U_0 \\ U_{-2} \\ U_2 \end{pmatrix} = \sum_{j=1}^{9} \begin{pmatrix} (r\, L_{j1}(\psi)) * W_j \\ (r\, L_{j2}(\psi)) * W_j \\ (r\, L_{j3}(\psi)) * W_j \end{pmatrix}. \tag{4.34}$$

As mentioned in Subsection 4.2.5 and Subsection 4.2.6, we expect that the ball saliency λ_2 will not give very useful information. Therefore, we are mainly interested in the resulting stick saliency map (i.e. $\lambda_1 - \lambda_2$) and the stick direction. Since this information is all contained in the $m = 2$ component ($\frac{1}{4}(\lambda_1 - \lambda_2) = |U_2|$ and $\angle \mathbf{e}_1 = \frac{1}{2} = \arg(U_2)$, Subsection 4.2.1) we can even further simplify steerable TV. For the $m = 2$ part of the voting field $\tilde{V}_2^\theta(r, \varphi)$ we find

$$\tilde{V}_2^\theta(r,\varphi) = \begin{pmatrix} e^{2i\theta} \\ 4 \\ 6e^{-2i\theta} \\ 4e^{-4i\theta} \\ e^{-6i\theta} \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} 1 \\ e^{2i\varphi} \\ e^{4i\varphi} \\ e^{6i\varphi} \\ e^{8i\varphi} \end{pmatrix} G_\sigma(r). \tag{4.35}$$

Now, we can implement the voting process as follows. As kernels, we need

$$\tilde{W}_m(r,\varphi) = e^{-\frac{r^2}{2\sigma_{\text{ctx}}^2}}\, e^{im\varphi} \quad \text{for } m = 0, 2, 4, 6, 8. \tag{4.36}$$

In cartesian coordinates this gives

$$W_m(x,y) = e^{-\frac{x^2+y^2}{2\sigma_{\text{ctx}}^2}} \left(\frac{x+iy}{|x+iy|}\right)^m \quad \text{for } m = 0, 2, 4, 6, 8, \tag{4.37}$$

see Figure 4.17. Given the local ridgeness r and orientation ψ, we need to calculate

$$C_m(x,y) = r(x,y)\, e^{-im\,\psi(x,y)} \quad \text{for } m = 0, 2, 4, 6. \tag{4.38}$$

Now, we can calculate the resulting $m = 2$ part by

$$U_2(x,y) = \begin{pmatrix} 1 \\ 4 \\ 6 \\ 4 \\ 1 \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} W_0 * \overline{C_2} \\ W_2 * C_0 \\ W_4 * C_2 \\ W_6 * C_4 \\ W_8 * C_6 \end{pmatrix}. \tag{4.39}$$

This gives for the stick map R and the orientation map Ψ

$$R(x,y) = |U_2(x,y)| \quad \text{and} \quad \Psi(x,y) = \frac{1}{2} \arg U_2(x,y). \tag{4.40}$$

This results in a formulation of TV that is *much more efficient* than would be expected at first sight.

The derivation above is for the case of $n = 2$ in Equation 4.25. We can do the same for other values of n, for $n = 1$ and $n = 3$ we derive

$$\text{for } n = 1: \quad U_2(x,y) = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} W_2 * C_0 \\ W_4 * C_2 \\ W_6 * C_4 \end{pmatrix}, \qquad \text{for } n = 3: \quad U_2(x,y) = \begin{pmatrix} 1 \\ 6 \\ 15 \\ 20 \\ 15 \\ 6 \\ 1 \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} \overline{W_2} * \overline{C_4} \\ W_0 * \overline{C_2} \\ W_2 * C_0 \\ W_4 * C_2 \\ W_6 * C_4 \\ W_8 * C_6 \\ W_{10} * C_8 \end{pmatrix}. \tag{4.41}$$

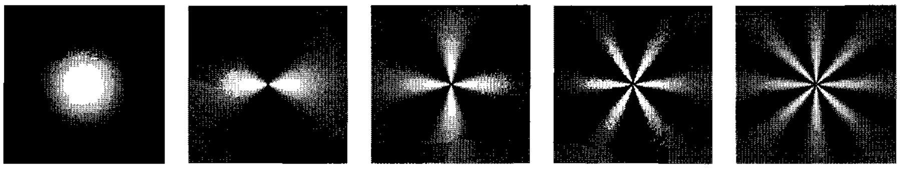


Figure 4.17: *Illustration of the real part of the basis functions W_m for $m = 0, 2, 4, 6, 8$. The imaginary parts are the same but with a phase shift in angle.*

4.4 Application to EP catheter images

In the previous section we derived a steerable TV method. In this section, we will describe the application of steerable TV on EP catheter images.

4.4.1 Examples on EP catheter images

To get an impression of the effect of TV, we show two examples on EP catheter images in Figure 4.18 and Figure 4.19. As input for these examples the Hessian filter is used (Chapter 3). For the voting process, the steerable filter with $n = 2$, as described in the previous sections, is applied. It is clear that the catheter is better visible in the context enhanced feature image than in the Hessian filter image. The "spaghetti-like" structures in the Hessian image are less visible in the voting result. This will make the extraction of the catheter based on the most salient line structures easier.

However, not only the catheter has become better visible, but also other structures in the image, for instance the ribs. This implies that other criteria are necessary to select the catheter. This is subject of Chapter 5. There are also some artefacts in the image. For instance, at the ablation tip the surrounding of the catheter looks very blurred. This is because the voting mechanisms does not take end-points (or in general features with an odd symmetry) into account.

If we look at the ball saliency map, it does not seem to contain useful information. Medioni et al. use this map to extract junctions. In Subsection 4.2.5 we already pointed out that this measure is not suitable in our case. In the second example, the junctions do give peaks in the ball saliency map, but this is not reliable enough for proper junction detection. The ball map also gives a high responses at the edges of the catheters. This is because at these locations, strong votes are received from all different orientations.

4.4.2 Repeated tensor voting

In principle, one could interpret the TV process as an advanced kind of blurring that takes local orientation into account. This means that if we would apply TV a few times consecutively, the feature images eventually will loose all structure.

Therefore, there is a need for a counteracting mechanism. In the voting result, quite some artefacts arise, which will be communicated again during a next voting step. Therefore, we want to extract the features that are most salient to belong to the line structure of interest, and discard uncertain features. In this way, in the next voting step, only certain features are communicated and enhanced again. The idea is schematically shown in Figure 4.20.

It is impossible to extract the salient features with 100% correctness. So, the voting step that follows the salient feature extraction should be able to fill small gaps, caused by removed information, and attenuate preserved information that actually does not belong to the features of interest.

We have considered two simple operations to extract salient features:

- A *threshold*, which for instance only keeps 20% of the most salient stick saliency tensors.
- *Thinning*: this means that only tensors are kept that are maximum relative to their neighbors in directions perpendicular to the local orientation. Details on the implementation of thinning can be found in Chapter 5 (Subsection 5.2.2).

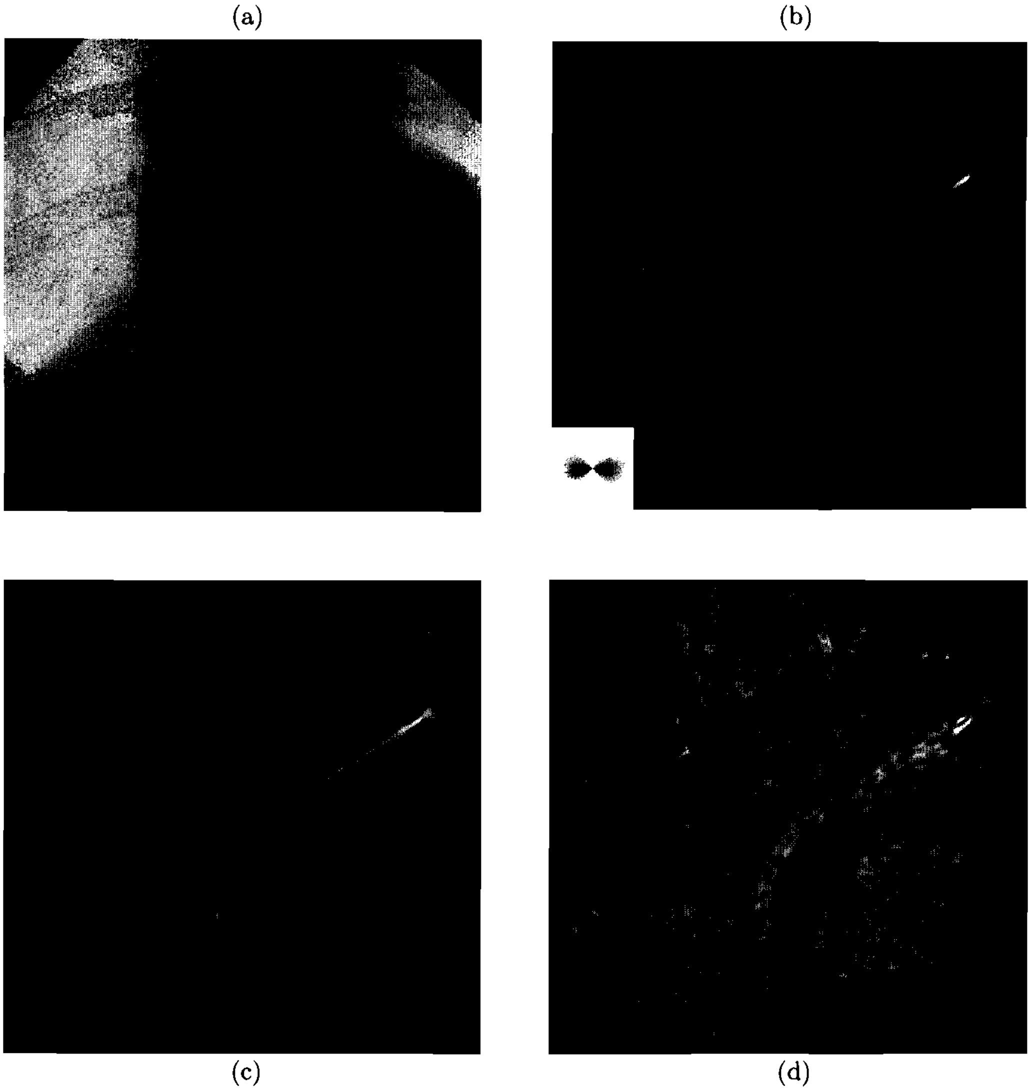


Figure 4.18: *Example of TV on a noisy EP image with a single ablation catheter. (a) Original image. (b) Hessian filter image, used as input. In the corner, the size of the voting field is shown. (c) Stick saliency map after voting. (d) Ball saliency map after voting.*

Figure 4.22 shows an example, in which TV is performed several times with thresholding (keeping 20% of the best saliency values) and thinning as intermediate processing steps.

We conclude that repeating TV one time, after thresholding and thinning, is useful, because it enhances the connectedness of the structures of interest. The threshold step can also be omitted, because it does not make much difference since thinning is the most important step for making the data sparse. Repeating TV several times proved to be not really useful, because the structures really get disturbed, especially the junctions and end-points, but also the localization of the line structures. Furthermore, applying TV several times makes the technique much more computational expensive. It becomes more like an iterative technique, but the iteration does not lead to a convergence. Note, that there are also iterative perceptual grouping techniques that do converge [17, Chapter 4]. Figure 4.21 shows the way we apply TV on EP catheter images.

The repeated TV scheme described above, shows some similarity with the iterative TV described in [11], with the major difference that Fischer et al. use *sparse* TV instead of dense TV.

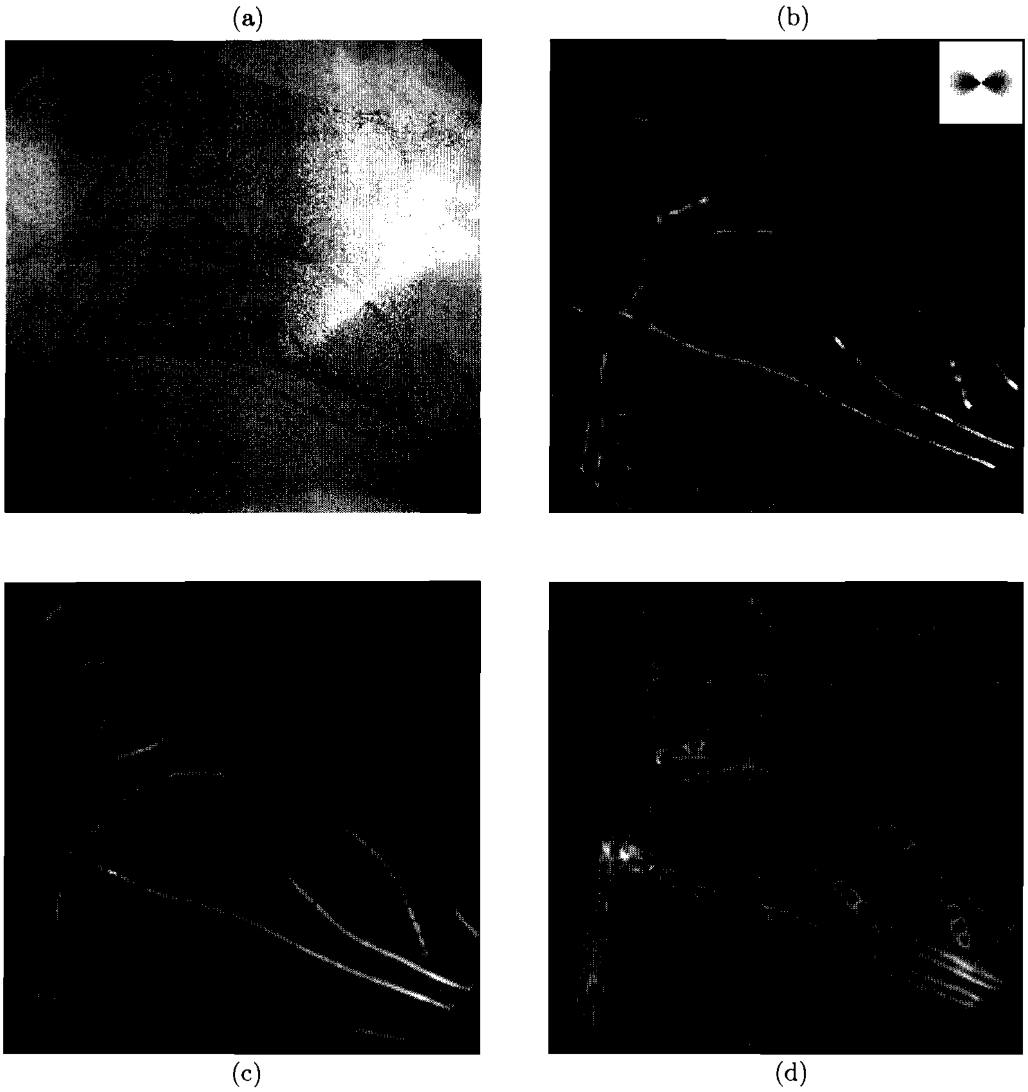


Figure 4.19: *Example of TV on a EP image with crossing catheters. (a) Original image. (b) Hessian filter image, used as input. In the corner, the size of the voting field is shown. (c) Stick saliency map after voting. (d) Ball saliency map after voting.*

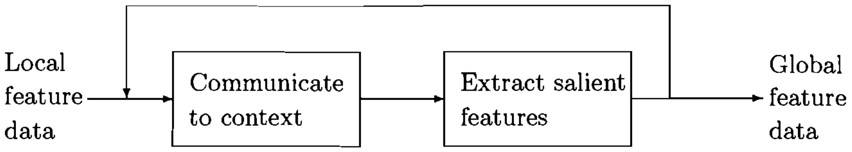


Figure 4.20: *Repeated TV: the output of context communication and extraction of salient features is fed back as input for a next iteration step.*

4.4.3 Context enhanced blobness

As shown in the black-box representation of Figure 4.1, also a context enhanced blobness measure is rendered. This measure is obtained by simply multiplying the local blobness with the context enhanced ridgeness $R(x, y)$

$$B(x, y) = b(x, y)\, R(x, y). \tag{4.42}$$

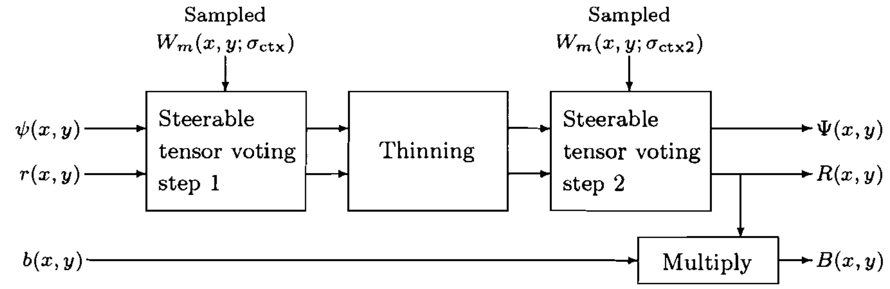


Figure 4.21: *Schematic overview of the context enhancement method for EP catheter images.*

The reason to do this is the knowledge that the electrodes are always situated on the catheter. Blobs that are not situated on line structures are weakened in this way.

4.5 Conclusions

In this chapter, we introduced the use of context filters, which are derived from the Gestalt principles, to enhance the EP catheters in a noisy feature image obtained with local filters. The reason to do this is to make the global search for the catheter in the image easier and more robust.

The most important source of inspiration is the tensor voting (TV) technique, as proposed by Medioni et al. [32]. We modified the technique in some points.

- Instead of the stick voting field as proposed by Medioni, we derived a steerable stick voting field.
- We applied the voting directly to a dense tensor field, consisting entirely of stick tensors.
- We decided not to use the second eigenvalue (ball saliency map) of the TV result.
- Using the above described modifications, we rewrote the TV operation as a linear superposition of a limited number of complex-valued convolutions.

The voting operation is expensive, because the kernels are quite large, but it can be implemented in parallel. In Section 6.3 the computational issues will be discussed in more detail.

There are some suggestions for improvements:

- Design of a better modelled voting field, using statistics or a more appropriate model of the EP catheters. However, the number of steerable components will probably increase.
- Include end-points in the voting scheme. In [38, 39] an extension to TV is proposed that also includes $|m| = 1$ tensor voting. In this approach, a vector field is calculated, called the *polarity field*. These vectors are sensitive to the direction from which the votes are received. So a large polarity vector is an indication for an end-point. If the polarity vector is zero, the same number of votes are received from both sides so we have an 180°-symmetry at that location.
- Include curvature in the voting scheme. We expect that if the voting field is adapted to the curvature, we obtain better context enhancement. In [36] an extension is proposed to measure the local curvature using TV.
- Some more sophisticated adaptations could be considered for the use of TV on gray level images instead of sparse tensor tokens. An interesting extension for this purpose is the use of inhibitory voting fields, proposed by Massad et al. [30, 31]. The idea is to first apply a non-maximum suppression method that is embedded in the TV framework by the use of inhibitory voting in the directions $\frac{\pi}{4} \leq \theta \leq \frac{3\pi}{4}$, followed by the normal excitatory voting in the directions $-\frac{\pi}{4} \leq \theta \leq \frac{\pi}{4}$.

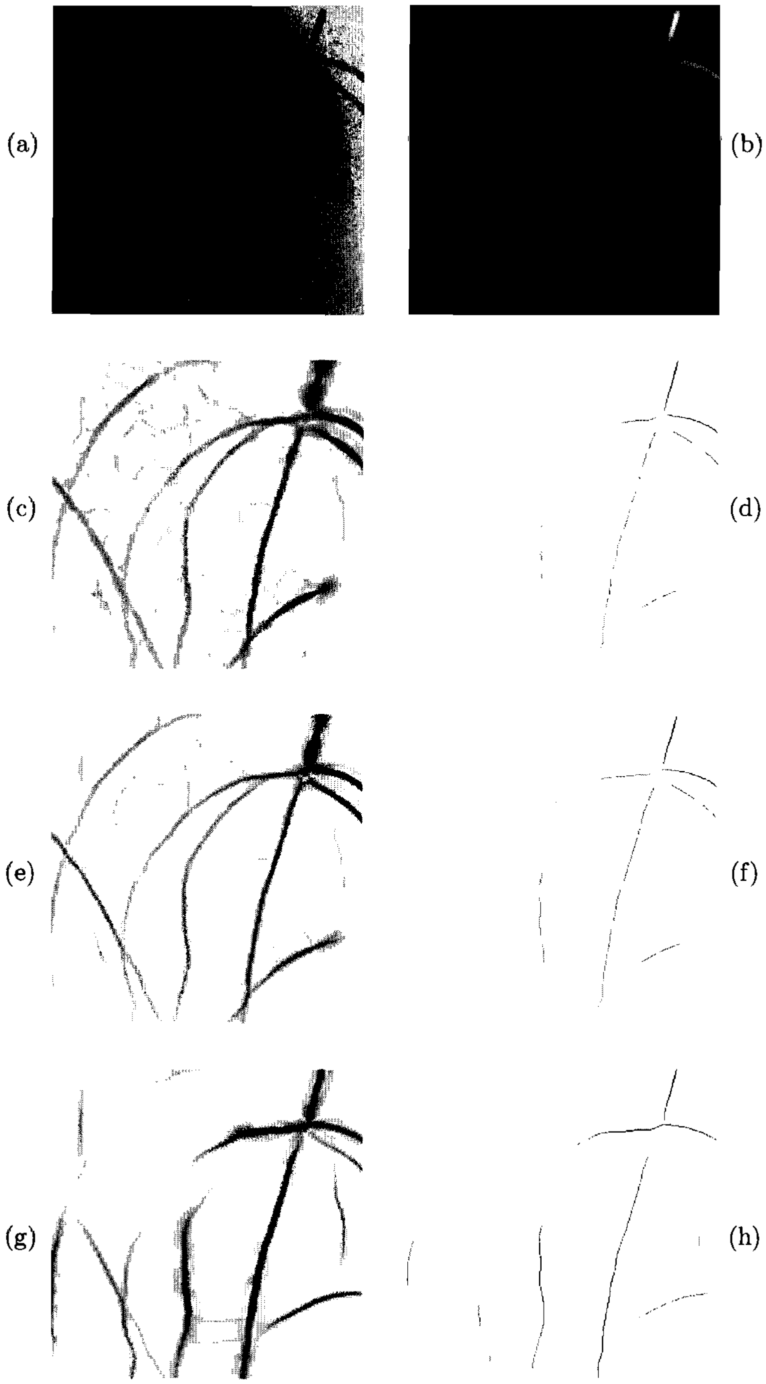


Figure 4.22: *Example of repeated TV. (a) Original image. (b) Hessian eigenvalue image, used as input for the first voting step. (c) Result after first TV and a threshold of 20%. (d) Result after thinning. (e) result after second TV step and threshold. (f) Result after thinning. (g) Result after 10 voting iterations. (h) Result after thinning. Note that images (c)-(h) have inverted colors for clarity.*

An interesting observation is the resemblance of our voting fields to magnetic and electric fields. For instance, the stick field shows similarity to the field lines of a magnetic dipole. The input tensors can be seen as a number of these dipoles, and the resulting field lines as the result of TV. Similarly, the ball field can be seen as an electric monopole. It could be interesting to further investigate the similarities between TV and electromagnetic fields. An interesting article for further inspiration is [21], where Jalba et al. propose a deformable model based on charged particles for segmentation.

Chapter 5

EP Catheter Extraction

In Chapter 3 and Chapter 4 we created feature images, in which the structures of interest are clearly visible, with minimal influence of noise. This chapter describes the decision step. With help of the generated feature images, we want to indicate where the EP catheter is located.

The black-box of the chapter is shown in Figure 5.1. As input, the context enhanced feature images of the previous chapter are used. As output, the catheter locations are returned. To accomplish this goal, we have to transform the pixel array images to a representation consisting of curves and points, of which the contextual relations are encoded in, for instance, *graphs*[1].

This chapter is built up as follows. In the first section, we will give an overview of the extraction method. In the next sections, the three main steps, i.e. *path extraction*, *electrode extraction and grouping*, and *path grouping*, are explained in detail. Next, the introduced parameters will be summarized and conclusions will be drawn.

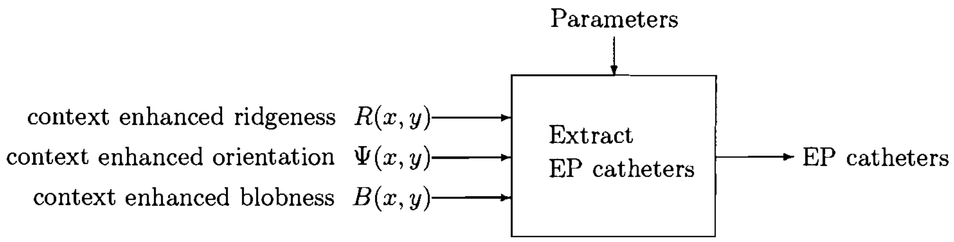


Figure 5.1: *Black-box of EP catheter extraction. As input, the context enhanced feature images obtained in Chapter 4 are used. The output is a collection of lines and points denoting the catheter locations and electrode locations respectively.*

5.1 Overview of the method

Figure 5.2 shows a scheme of the basic steps of the method. There are three main blocks, indicated by the dotted boxes:

- *Path extraction* (Section 5.2). This block extracts the most salient line segments in the ridgeness image R and finds spatial relations between them, resulting in a *path graph* consisting of *paths* and *connections* between paths. A path is a trajectory in the image (defined in Subsection 5.2.1). A single extracted path usually does not represent a single object (e.g. a catheter) in the image, because the path extraction algorithm does not handle special points in the image, like crossings of two lines. At the same time, a single path should always be a part of a *single object*, i.e. a path should not encompass parts of two different objects in the image. A connection between paths indicates that the paths *potentially* belong to the same

---

[1] There are several meanings of the word "graph". In this chapter, we mean a graph that consists of a set of *nodes*, and a set of *edges* connecting a subset of the nodes

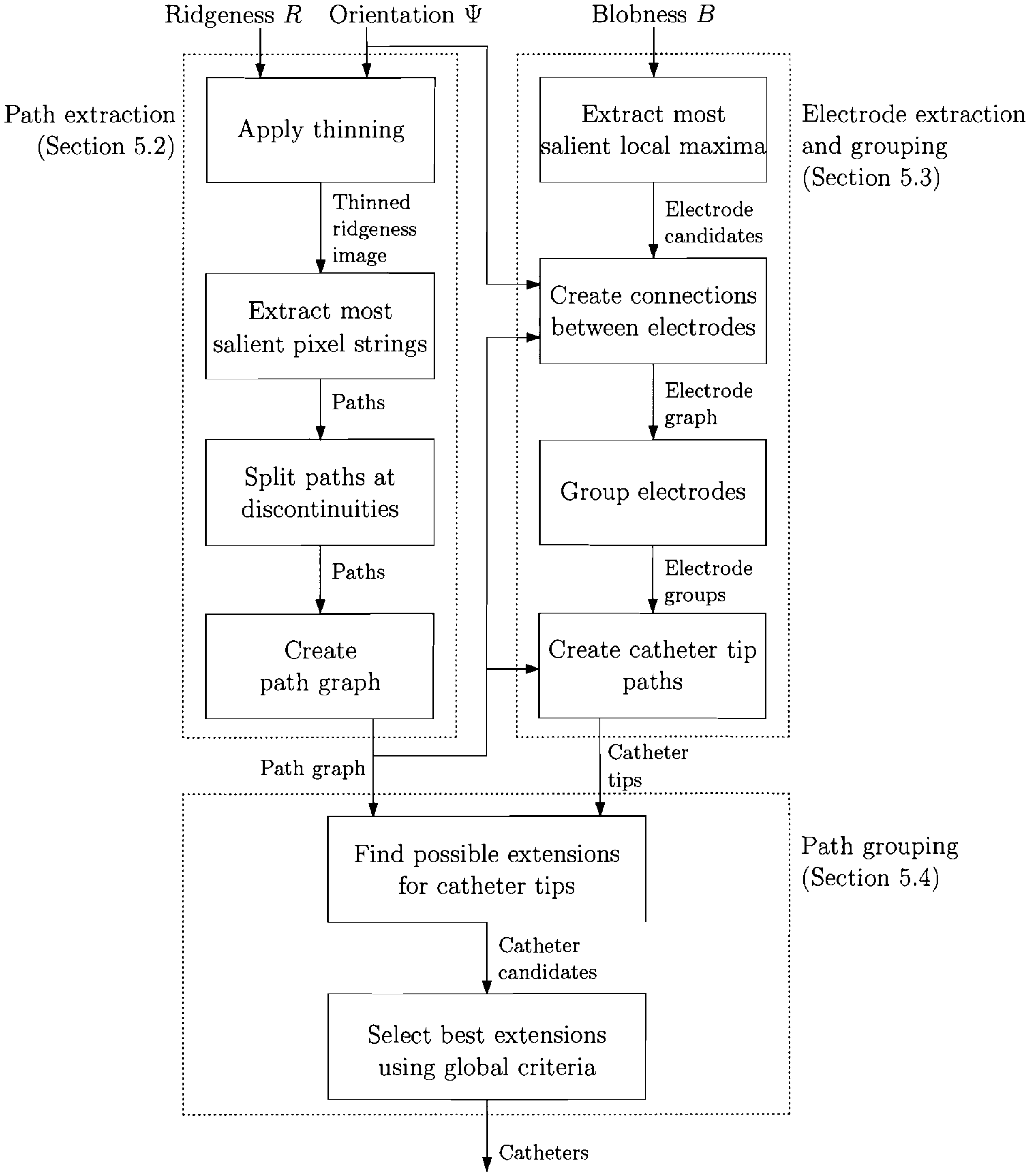


Figure 5.2: *Schematic overview of the EP catheter extraction method.*

object. The created connections between paths in the path graph are the first preparation step for further grouping of the paths.

The block consists of four steps. The *thinning* step manipulates the ridgeness image R such that only ridge maxima are retained (Subsection 5.2.2). Next, the most salient *pixel strings* are extracted and stored as paths (Subsection 5.2.3). Then, some paths that could be part of branching lines are split to allow for a proper reconnection in the subsequent step (Subsections 5.2.4 and 5.2.5). Finally, a *path graph* is created, which has connections between paths that could potentially belong to the same global object (Subsection 5.2.6).

- *Electrode extraction and grouping* (Section 5.3). This block searches for groups of blobs in the image that have a spatial configuration typical for EP catheter electrodes. The electrodes offer useful information for finding the tip. Each extracted electrode group should represent the electrodes situated on a single EP catheter in the image.

  The block consists of four steps. The first step extracts the most salient blob responses from the blobness image B (Subsection 5.3.1). The resulting set of *electrode candidates* are the nodes of an *electrode graph* (Subsection 5.3.2). This graph has connections between electrode candidates if their spatial relation indicates that they are two neighboring electrodes on an EP catheter. For this step the information about orientation and the location of paths is also used. In the next step, the *grouping of electrodes*, we scan for groups of connected candidates in the electrode graph that have the best match with the known properties of electrodes on an EP catheter. Finally, *catheter tip paths* are created (Subsection 5.3.4) that connect all electrodes of an electrode group with each other.

- *Path grouping* (Section 5.4). The goal of this block is to group the paths into longer paths that form the EP catheters. The extracted catheter tips are suitable as starting points (seeds) for this grouping process. The tip paths are extended until a certain stopping criterion is fulfilled.

  Similarly to the electrode grouping process, first a number of extension candidates are created (Subsection 5.4.1). Then, using knowledge on the global properties of an EP catheter, the most likely candidates are selected (Subsection 5.4.2). Ideally, the selected candidates match the catheters in the image.

The extraction of EP catheters is a difficult problem. It is unavoidable that a lot of parameters, like thresholds, are introduced. The goal is to design the algorithm in such a way that each single parameter is not volatile. For instance, a threshold that restricts the number of possible solutions in an intermediate step should be chosen in a permissive way, because it is better to keep some false solutions instead of discarding true solutions. The introduced parameters in this chapter are indicated using `typewriter font`. For clarity, the prefix tells something about the quantity of the parameter, see Table 5.1.

Name prefix	*Type of parameter*	*Parameter domain*
`n_`...	Number of ...	$p \in \mathbb{Z}^*$ (nonnegative integer)
`d_`...	Distance of ... (in pixels)	$p \in \mathbb{R}$ and $p \geq 0$
`ca_`...	Cosine of angle of ...	$p \in [-1, 1]$
`r_`...	Ratio of ...	$p \in \mathbb{R}$ and $p \geq 0$

Table 5.1: *Prefixes for different types of parameters. The last column specifies the domain for each class of parameter.*

Some steps of the presented method show resemblance with the methods designed by Philips Research, Paris, for the detection of guide wires [6, 13]. Especially the part that extracts the paths is inspired on that work. However, we could not really compare our work with the work in Paris, because the information we had available was not detailed enough.

In the rest of this chapter, all steps will be described in detail. We will use one EP image to demonstrate the algorithm. The original image and feature images are shown in Figure 5.3.

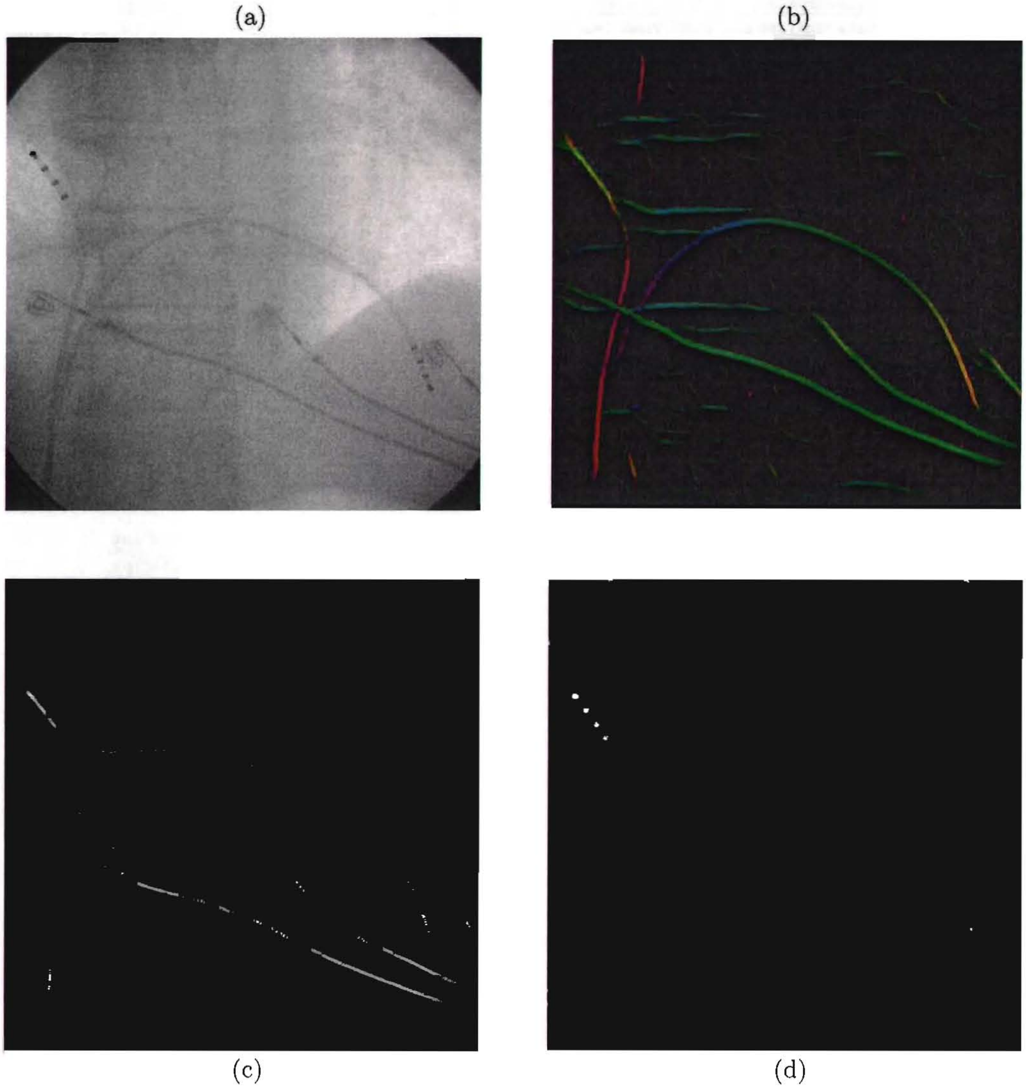


Figure 5.3: *EP catheter example image used throughout this chapter. (a) Original image. (b) Context enhanced orientation, encoded by color. (c) Context enhanced ridgeness image. (d) Blobness image.*

5.2 Path extraction

This section describes our methods to extract the most salient line structures and represent them as so-called *paths*. In the first subsection, the way we represent these paths is defined. Then, we describe the algorithmic steps that are performed (see Figure 5.2).

5.2.1 Path representation

A *path* is used to represent centerlines of line structures extracted from an image. It is called path because it is a trajectory through the image domain. In this work, paths are represented as a list of neighboring pixel coordinates:

$$\mathcal{P} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n\}, \tag{5.1}$$

where $n = |\mathcal{P}|$ is the number of pixel coordinates, and $\mathbf{p}_i$, $1 \leq i \leq n$, are two-dimensional integer coordinates on the pixel grid. A valid path must fulfill the following properties:

- $\mathbf{p}_i \in \Omega$, for all $1 \leq i \leq n$.

- The list has an ordering, such that $\mathbf{p}_i$ and $\mathbf{p}_{i+1}$ are 8-connected neighbors of each other, i.e. $0 < ||\mathbf{p}_i - \mathbf{p}_{i+1}|| \leq \sqrt{2} \quad \forall\, 1 \leq i < n$.
- $\mathbf{p}_i \neq \mathbf{p}_j$ for all $i \neq j$ and $1 \leq i, j \leq n$.

The *reversed path* of path $\mathcal{P}$, denoted $\check{\mathcal{P}}$, is defined as:

$$\check{\mathcal{P}} = \{\mathbf{p}_n, \mathbf{p}_{n-1}, \ldots, \mathbf{p}_1\}. \tag{5.2}$$

The *saliency* of a path $s(\mathcal{P})$ is defined as the cumulated ridgeness of the path:

$$s(\mathcal{P}) = \sum_{i=1}^{n} R(\mathbf{p}_i). \tag{5.3}$$

The *set of all paths* that are extracted from the image is denoted by $\mathcal{Q}$

$$\mathcal{Q} = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_{|\mathcal{Q}|}\}. \tag{5.4}$$

The list-of-coordinates representation is chosen because it is the most straightforward and simple one to implement. Different representations are possible:

- We could allow non-integer coordinates in the path description, to get a smoother path description and sub-pixel accuracy. However, we do not need sub-pixel accuracy for the application, because in the end we are interested in the *pixels* that belong to the catheter, for instance for the measurement of the contrast-to-noise ratio from the EP catheter and its background.
- Another interesting representation is a spline, which enables us to describe the path as a mathematical function. The advantage is that some calculations are easier to perform. We did not choose it due to time limitations of this project, and because for this application it might be a detour.

5.2.2 Thinning of ridgeness

The intention of the thinning operation is to only keep the centerlines of ridges. This step is a necessary preprocessing step for path extraction. The desirable output of the thinning step is a sparse pixel map (i.e. most pixels must be zero), which only contains *pixel strings*. A pixel string is defined as a group of nonzero pixels, where each pixel has exactly 2 neighboring nonzero pixels except for two of these pixels: these two pixels are the end-points and thus only have 1 nonzero neighbor. We will call this the *pixel string constraint*. See also [13].

Given a ridgeness feature image $R(x, y)$, and orientation image $\Psi(x, y)$, with $(x, y) \in \Omega$, the thinning operation is defined as

$$\text{Thinning}[R, \Psi](x, y) = \begin{cases} R(x, y) & \text{if } R(x, y) > R(\text{neighbor}_1) \text{ and } R(x, y) > R(\text{neighbor}_2); \\ 0 & \text{otherwise,} \end{cases} \tag{5.5}$$

where neighbor_i are the two pixel neighbors in the orientation perpendicular to the orientation specified by $\Psi(x, y)$, i.e.

$$\begin{aligned} \text{neighbor}_1 &= (x + \text{cround}[-\sin \Psi(x, y)], y + \text{cround}[\cos \Psi(x, y)]), \\ \text{neighbor}_2 &= (x - \text{cround}[-\sin \Psi(x, y)], y - \text{cround}[\cos \Psi(x, y)]), \end{aligned} \tag{5.6}$$

with

$$\text{cround}(x) = \begin{cases} -1 & \text{for } x \leq -1/\sqrt{2}; \\ 0 & \text{for } -1/\sqrt{2} < x < 1/\sqrt{2}; \\ 1 & \text{for } x \geq 1/\sqrt{2}. \end{cases} \tag{5.7}$$

A few notes should be made:

- The angles Ψ are rounded to fit on the pixel grid. This means there are only 4 distinct orientations. This can lead to errors. It is possible to interpolate values for higher precision. However, it is expected that this will not make much difference, because the input data has a certain smoothness due to the local filtering and tensor voting step.
- In the implementation, it is not necessary to calculate the cosine and sine of Equation 5.6. We can just round off the orientation and then check the 2 corresponding adjacent pixels.
- In the implementation, it is necessary to check for the image border.

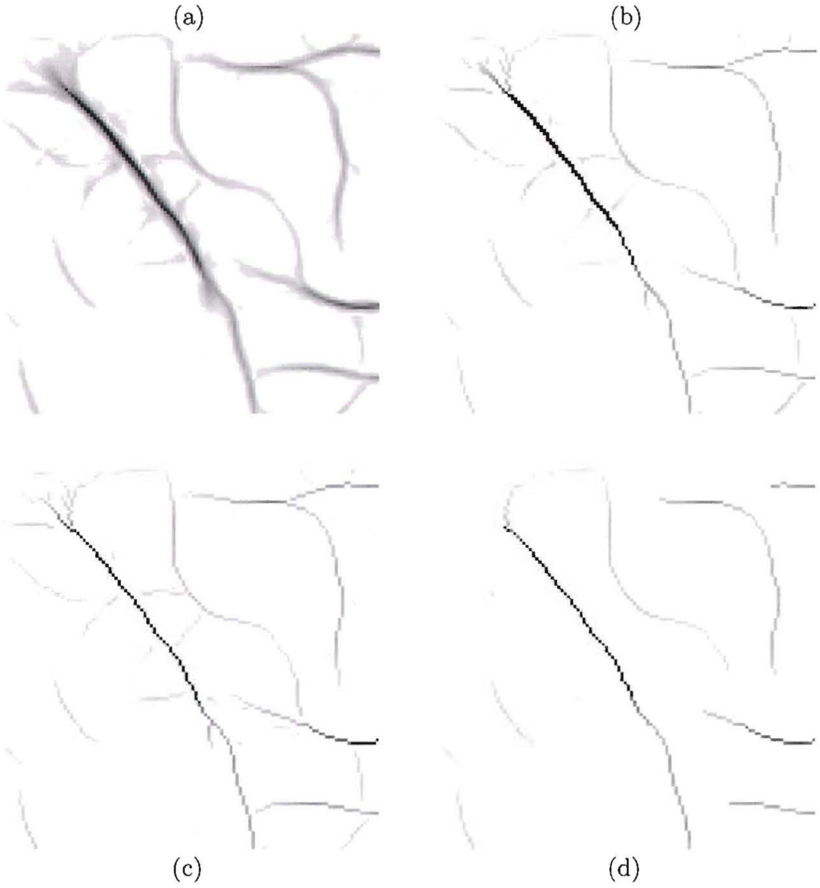


Figure 5.4: *Example of thinning. (a) Input image R (result of tensor voting). (b) Result after thinning. Only the centerlines are kept. (c) Result after skeletization. (d) Result after 14 pruning steps. The colors of all images are inverted.*

An example of thinning is shown in Figure 5.4a (input image) and Figure 5.4b (result after thinning). The result is a sparse image, in which the majority of the pixels is zero.

Note that there are pixels that have more than 2 nonzero neighbors, which means that the result of thinning does not fulfill the pixel string constraint. If the catheter has an orientation of about 45° (mod 90°), the thinned image shows a line that is two pixels wide. To solve this problem, a single *skeletization* step has to be performed. Skeletization is performed by matching the patterns of Figure 5.5a-b in 4 different orientations in the thinned image, for every 3×3 pixel neighborhood. If a match occurs, the pixel in the center of the 3×3 neighborhood is removed[2]. The result of this operation is shown in Figure 5.4c.

---

[2] Note, that this method is not independent of the order, in which the image is scanned. If the image is scanned from top to bottom, we obtain different results than if the image is scanned from bottom to top. However, this problem will only lead to small deviations of the extracted centerlines, so it is not considered a major problem.

After skeletization it is still not guaranteed that all nonzero pixels fulfill the pixel string constraint. It is possible that a pixel has for instance 3 nonzero neighbors. The pixel string extraction algorithm should take this into account.

Sometimes, the thinning operation creates short terminal strings, a kind of "pigtails". To remove these artefacts, a number of *pruning* steps (`n_PruningSteps`) can be performed. Pruning removes pixels that match the patterns of Figure 5.5c-d. This operation must be handled with care, because it removes pixels at all end-points of pixel strings. However, since tensor voting has the tendency to make line structures in images longer, it is not a problem if a few end-point pixels are removed.

(a)

0	0	0
X	1	X
1	1	1

(b)

X	0	0
1	1	0
X	1	X

(c)

0	0	0
0	1	0
X	X	0

(d)

0	0	0
0	1	0
0	X	X

Figure 5.5: 3×3 *pixel patterns for skeletization and pruning. These patterns are matched in the thinned image, and if a match is found, the pixel at that location is made zero. The meaning of the symbols is: 0 = zero pixel, 1 = nonzero pixel, X = don't care about the value of the pixel. All four patterns must be used with rotations of* 0°, 90°, 180°, *and* 270°. *(a) and (b) Patterns used for skeletization. (c) and (d) Patterns used for pruning.*

5.2.3 Extraction of pixel strings

The intention of this step is to extract a number of (`n_PathExtract`) most salient paths. As input, the thinned and skeletonized ridgeness image $R_{\mathrm{T}}(x, y)$, consisting of pixel strings, is used. The algorithm works as follows.

```
Algorithm 2: Extraction of most salient paths
Repeat
    • Search the pixel in feature image $R_T$ with highest value. This pixel is used as seed.
    • Extract the entire pixel string to which the selected pixel belongs by propagating
      to both sides starting at the seed. If a pixel is found with a single neighbor, the
      propagation stops and the pixel string is entirely extracted. If a pixel is found with
      more than 2 neighbors, the pixel string constraint is violated and the propagation of
      the current path is stopped.
    • Remove the pixels belonging to the extracted path from $R_T$.
    • If the resulting path $\mathcal{P}$ is too small, $|\mathcal{P}| \leq$ th_MinPathLength, discard the path, otherwise keep it.
Until n_PathExtract paths are extracted.
```

Figure 5.6 gives an example of the path extraction step.

The thinning and pixel string extraction steps can not always handle locations where line structures come together appropriately. Crossings and branches of line structures are most problematic, see Figure 5.7. In the next steps, *electrode extraction* and path grouping, we will extract the EP catheters by concatenating paths. This will fail, however, if situations as in Figure 5.7 occur. Therefore, there is a need to split paths in these situations. If there is ambiguity whether a path should be split or not, it is better to split the path, because the subsequent steps can reconnect the paths if necessary. We consider two criteria to split a path:

- Discontinuities in the path. If the path has a strong discontinuity, the saliency that this path belongs to one structure is small.
- The path is close to an end-point of another path. In this case it makes sense to split the path at the position that is closest to the end-point of the other path.

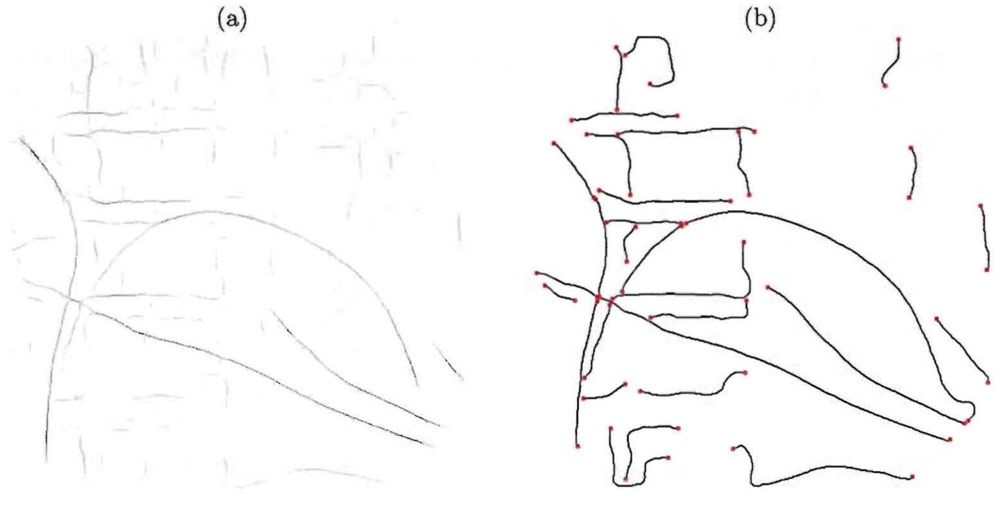


Figure 5.6: *Example of path extraction. (a) Thinned image used as input for path extraction (gray values are reversed). (b) Result after extracting the 20 most salient paths. The red circles indicate the end-points of the paths.*

Note, that the two mentioned reasons to split a path in practice often coincide. This is also the case in Figure 5.7a for the most left split point. The algorithms to split the paths are subject of the next two subsections.

5.2.4 Splitting paths at discontinuities

To detect discontinuities in a path $\mathcal{P}$, we need a measure for the curvature. If the curvature exceeds a certain threshold, the path should be split. Mathematically, the direction of a parametric curve $\mathbf{f}(t)$ is simply its first derivative $\dot{\mathbf{f}}(t)$, and its curvature is the second derivative $\ddot{\mathbf{f}}(t)$. The paths are defined on an integer coordinate grid, which means we need to perform smoothing during direction estimation.

A rough estimation of direction at a coordinate $\mathbf{p}_i$ is the difference vector $\mathbf{p}_i - \mathbf{p}_{i-1}$. This estimation only gives 8 possible directions. For practical reasons, to keep the implementation simple and efficient, we choose a first-order recursive filter for the estimation of directions and curvature. The impulse response of this filter is

$$h(t) = \begin{cases} (\alpha_\mathrm{d})^t & \text{if } t \geq 0; \\ 0 & \text{if } t < 0, \end{cases} \tag{5.8}$$

where α_d is the exponential decay factor. This is a causal filter that we apply in the causal direction, i.e. from left to right in the sequence of coordinates in path $\mathcal{P}$, *and* in the anti-causal direction, i.e. from right to left in the list $\mathcal{P}$. This gives the following recursive definitions of the left-to-right and right-to-left filter

$$\begin{aligned} \mathbf{d}_\mathrm{L}[1] &= (0,0) \\ \mathbf{d}_\mathrm{L}[i] &= (\mathbf{p}_i - \mathbf{p}_{i-1}) + \alpha_\mathrm{d}\,\mathbf{d}_\mathrm{L}[i-1], \quad 2 \leq i \leq n, \end{aligned} \tag{5.9}$$

$$\begin{aligned} \mathbf{d}_\mathrm{R}[n] &= (0,0) \\ \mathbf{d}_\mathrm{R}[i] &= (\mathbf{p}_i - \mathbf{p}_{i+1}) + \alpha_\mathrm{d}\,\mathbf{d}_\mathrm{R}[i+1], \quad 1 \leq i < n, \end{aligned} \tag{5.10}$$

where $n = |\mathcal{P}|$, and $\mathbf{d}_\mathrm{L}[i]$ and $\mathbf{d}_\mathrm{R}[i]$ are the responses of the left-to-right resp right-to-left recursive filter at path position i. The initialization $(0,0)$ is chosen because "in the beginning" we do not know the direction. Figure 5.8 shows the scheme of this filter. The transfer function of this digital filter is given by $H(z) = \frac{z-1}{z-\alpha_\mathrm{d}}$, so the filter has a zero for $z = 1$ and a pole for $z = \alpha_\mathrm{d}$.

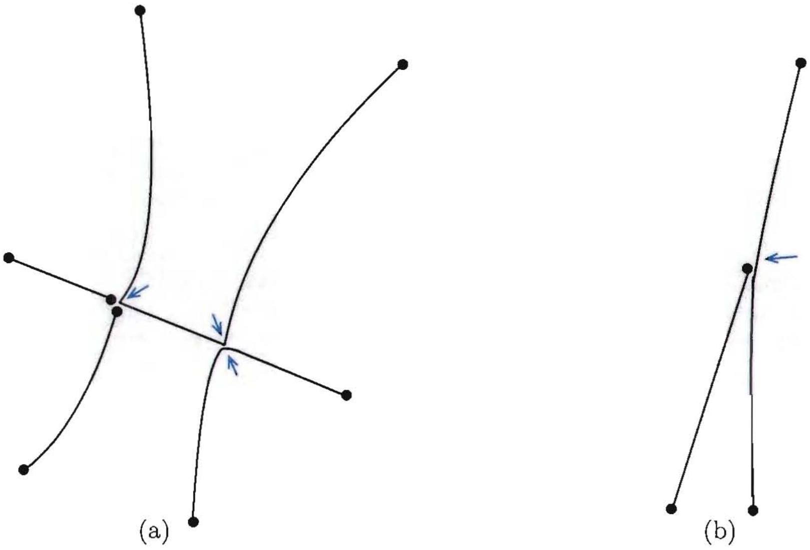


Figure 5.7: *Two examples of path configurations, in which paths should be split. The small circles indicate end-points of paths, and the blue arrows indicate the split positions. (a) Example with crossings of line structures. (b) Example with a branch of line structures. This case is more ambiguous, but it is better to split the path because subsequent steps can reconnect the paths if necessary.*

With this left-to-right and right-to-left filtering method, we have for every pixel position on the path a *left-direction* estimation, in which only information on the left side is used, and a *right-direction* estimation, in which only information on the right side is used. The resulting directions are

$$\theta_{\mathrm{L}}[i] = \angle(\mathbf{d}_{\mathrm{L}}[i]) \quad \text{and} \quad \theta_{\mathrm{R}}[i] = \angle(-\mathbf{d}_{\mathrm{R}}[i]). \tag{5.11}$$

The *difference* between the left-direction and the right-direction

$$\rho[i] = \theta_{\mathrm{L}}[i] - \theta_{\mathrm{R}}[i] \pmod{2\pi}, \tag{5.12}$$

is an indication for *curvature* at position i. The path positions with high curvature are detected by finding local maxima of $|\rho|$. So if we have a maximum at position j, i.e.

$$|\rho[j]| > |\rho[j+k]| \text{ for all } k \in \{-2,-1,1,2\}, \tag{5.13}$$

and the curvature at this local maximum is above a certain threshold

$$|\rho[j]| > \arccos(\texttt{ca\_CurvatureSplit}), \tag{5.14}$$

then we have a discontinuity, and the path $\mathcal{P}$ is split in two paths at position j.

In the implementation, we can omit the use of trigonometric functions by rewriting Equation 5.14 as

$$\frac{-\mathbf{d}_{\mathrm{L}}[i]\cdot\mathbf{d}_{\mathrm{R}}[i]}{||\mathbf{d}_{\mathrm{L}}[i]||\ ||\mathbf{d}_{\mathrm{R}}[i]||} = \cos|\rho[j]| < \texttt{ca\_CurvatureSplit}. \tag{5.15}$$

Some examples are shown in Figure 5.9. In the first example, a straight line, it can be seen that the measured right-direction and left-direction are nearly the same everywhere, implying that the curvature is low. In the second case, a path that has the shape of a sine, we can see clear peaks with opposite signs. In the last case, we can again see a clear peak. In all cases, the direction of the end-points are estimated reasonably well. In Figure 5.10b an example is shown on real EP catheter path data.

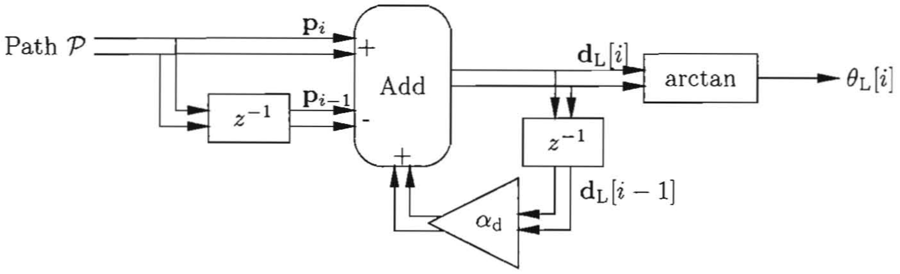


Figure 5.8: *Scheme of the first-order digital IIR filter used for the estimation of directions. The double arrows indicate vector-valued signals.* z^{-1} *indicates a memory (delay of 1 discrete time step) and the triangle indicates a multiplicator. For the right-direction estimation* θ_R*, the reversed path* $\check{\mathcal{P}}$ *is used as input instead of* $\mathcal{P}$*.*

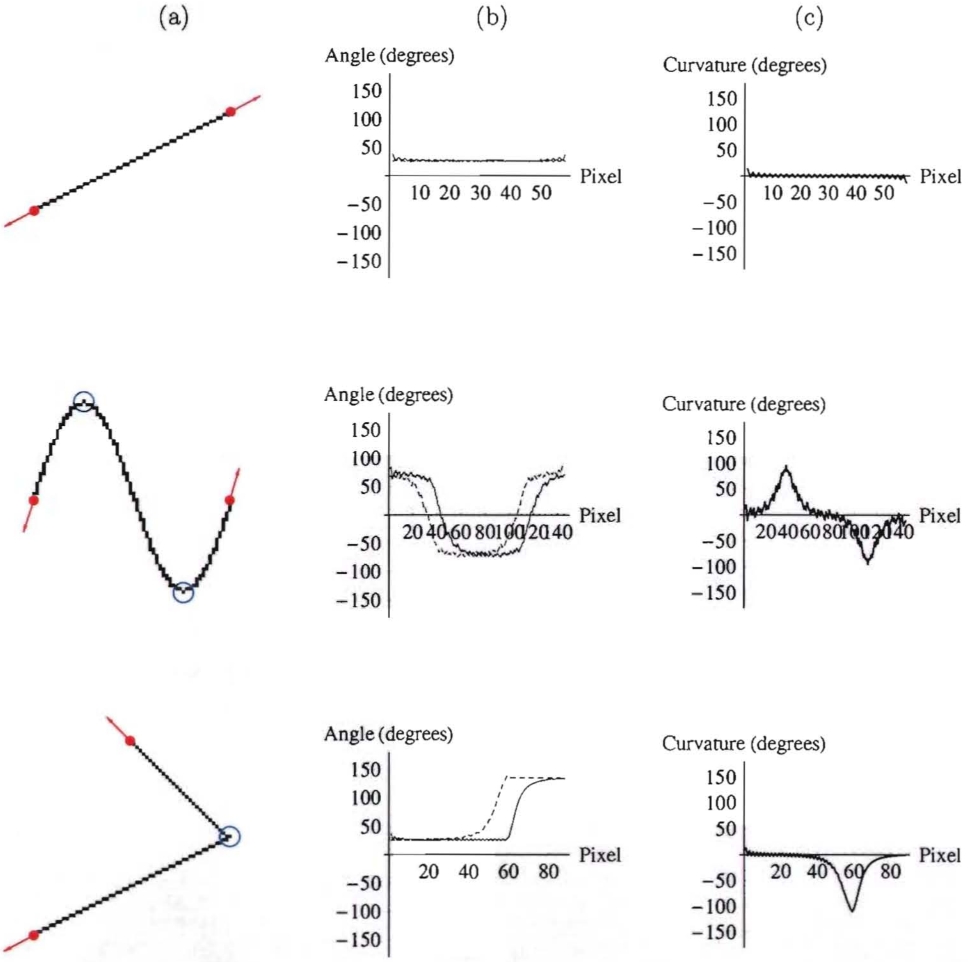


Figure 5.9: *Examples of direction and curvature estimation. Column (a): Pixel paths, the red arrows at the end-points indicate the estimated end-point directions. The blue circles indicate points with high curvature. Column (b): Estimated angles following from left-to-right direction estimation (solid line) and right-to-left direction estimation (dashed line). The left side of each path corresponds to the left-most end-point of the visualized path. Column (c): The difference of estimated left-to-right and right-to-left directions give an indication for curvature, and give strong peaks at discontinuities. In this examples,* $\alpha_d = 0.88$*, and* ca_CurvatureSplit $= \pi/3$*.*

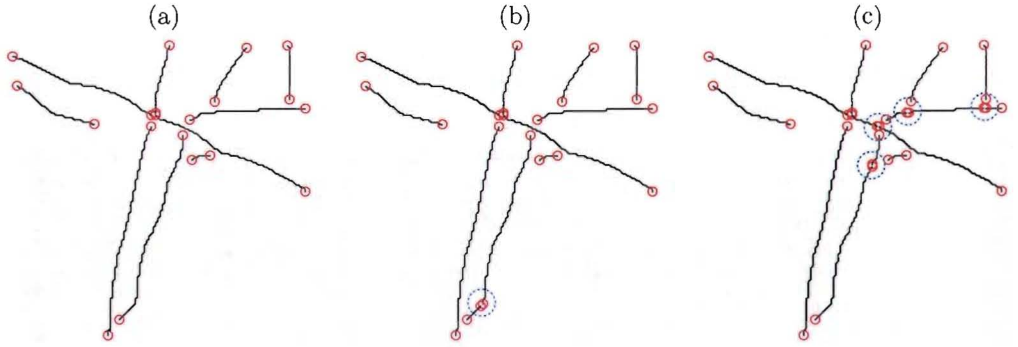


Figure 5.10: *Example of path splitting. (a) Path extraction without any splitting (this is a part of Figure 5.6). (b) Paths are split at discontinuities (Subsection 5.2.4). The blue dashed circle indicates the change. (c) Paths are split at collisions with end-points (Subsection 5.2.5). Again, the blue dashed circles indicates the changes.*

5.2.5 Splitting paths at collisions with end-points

When a path $\mathcal{P}_1$ comes close to an end-point of another path $\mathcal{P}_2$, it can happen that one segment of path $\mathcal{P}_1$ should be grouped with $\mathcal{P}_2$. To allow proper reconnection we split a path at a position that is too close to the end-point of the other path.

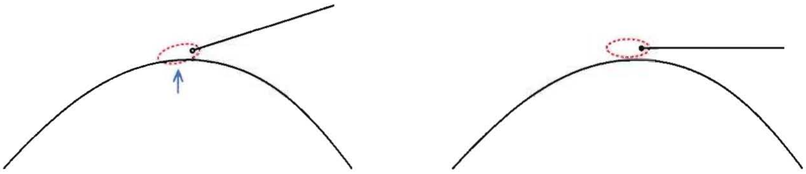


Figure 5.11: *Example of splitting a path at a collision with an end-point. Although the Euclidean distance between end-point and path is the same, it is more appropriate to split the path in the left example, than in the right example. The dashed ellipse shows the region around the end-point that infers a path split.*

Figure 5.11 shows the importance of directional information at the end-points, implying there should be anisotropy in the distance measure, which depends on the direction of the end-points. For the estimation of the direction of the left end-point we can use the right-direction at $i = 1$, $\theta_\mathrm{R}[1]$, and for the right end-point the left-direction $\theta_\mathrm{L}[n]$, according to the direction estimation method described in the previous subsection. We will further on denote the directions at the two end-points of a path $\mathcal{P}$ as

$$\theta_{\mathrm{E1}}(\mathcal{P}) = \theta_\mathrm{R}[1] \quad \text{and} \quad \theta_{\mathrm{E2}}(\mathcal{P}) = \theta_\mathrm{L}[|\mathcal{P}|]. \tag{5.16}$$

To split paths using end-point information we first create an image, in which we draw for each end-point a short line segment of length `d_PathEndptAnisotropy` in the estimated direction of the end-point. On this image we apply the *Euclidean distance transform* as described in [7]. The resulting distance map contains for every pixel position (x, y) the distance to the closest end-point, where `d_PathEndptAnisotropy` determines the amount of anisotropy in the distance. A path is split if it draws nearer than a certain distance (`d_PathEndptSplit`) to an end-point.

Algorithm 3: *Split paths at collision with end-points*

- Create an end-point distance map:
  - Create an empty pixel map $D_{\text{endpt}}(x, y)$
  - For all end-points of all paths in $\mathcal{Q}$, draw a line segment of length `d_PathEndptAnisotropy` in the estimated direction of the end-point in pixel map D_{endpt}.
  - Apply Euclidean distance transform to D_{endpt}. Now, D_{endpt} contains for every pixel position (x, y), the minimum Euclidean distance to a line segment belonging to an end-point.
- Search for paths drawing near to end-points.
  For all paths in $\mathcal{Q}$:
  - Iterate through all pixel positions $\mathbf{p}_i$ in the current path. If $D_{\text{endpt}}(\mathbf{p}_i) \leq$ `d_PathEndptSplit` and $D_{\text{endpt}}(\mathbf{p}_i) < D_{\text{endpt}}(\mathbf{p}_{i-1})$ and $D_{\text{endpt}}(\mathbf{p}_i) < D_{\text{endpt}}(\mathbf{p}_{i+1})$: split the path at position i.

An example is shown in Figure 5.10c.

5.2.6 Creation of path graphs

The path graph takes the paths in $\mathcal{Q}$ (the set of all extracted paths) as nodes. Non-directional connections (edges) are created between paths that potentially should be connected (i.e. that could belong to the same global structure)[3]. The set of all connections between paths is denoted as $\mathcal{G}$. For example, $\mathcal{G} = \{\{\mathcal{P}_i, \mathcal{P}_j\}, \{\mathcal{P}_k, \mathcal{P}_l\}\}$ means that paths $\mathcal{P}_i$ and $\mathcal{P}_j$ are connected, as well as $\mathcal{P}_k$ and $\mathcal{P}_l$. Note that $\{\mathcal{P}_i, \mathcal{P}_j\}$ is equivalent to $\{\mathcal{P}_j, \mathcal{P}_i\}$. To decide whether two paths should be connected in the path graph, we use the positions and the directions of the end-points of the paths.

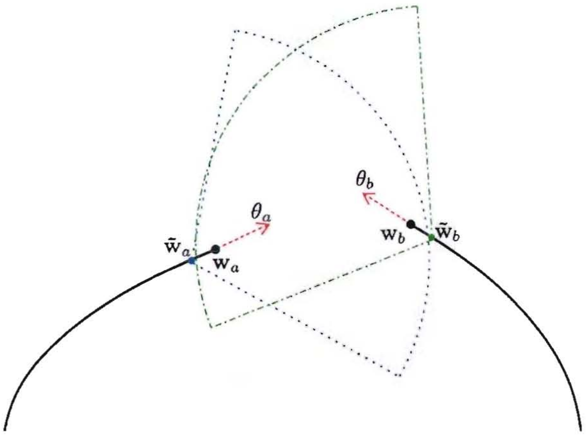


Figure 5.12: *Illustration of the path connection criterion. The end-points form a potential connection if and only if end-point $\mathbf{w}_b$ falls in the wedge-shaped region of end-points $\mathbf{w}_a$ (indicated by the blue dotted line),* and *end-point $\mathbf{w}_a$ falls in the wedge-shaped region of end-points $\mathbf{w}_b$ (indicated by the green dashed-dotted line). The corners of the wedge-shaped regions, indicated by $\tilde{\mathbf{w}}_a$ and $\tilde{\mathbf{w}}_b$, are shifted over a small distance to achieve less directional sensitivity for end-points that are really close to each other.*

Suppose we have an end-point with coordinate $\mathbf{w}_a$ and an end-point $\mathbf{w}_b$, belonging to two *different* paths, with directions θ_a and θ_b. The paths with these end-points are connected if the

---

[3] Another possible method, which we considered earlier, is to create a path-junction-graph by explicitly detecting crossings (junctions) between line structures. However, it was hard to detect junctions robustly, and the approach is rather error-prone. Because we are not primarily interested in where the junctions are, but in the curves that make the EP catheters, this approach was discarded.

following condition holds

$$\begin{aligned} & ||\mathbf{w}_a - \mathbf{w}_b|| \leq \texttt{d\_MaxInterEndptDistance} \\ \wedge\ & \mathbf{d}_a \cdot \frac{(\mathbf{w}_b - \tilde{\mathbf{w}}_a)}{||\mathbf{w}_b - \tilde{\mathbf{w}}_a||} \geq \texttt{ca\_PathConnectRegionAngle} \\ \wedge\ & \mathbf{d}_b \cdot \frac{(\mathbf{w}_a - \tilde{\mathbf{w}}_b)}{||\mathbf{w}_a - \tilde{\mathbf{w}}_b||} \geq \texttt{ca\_PathConnectRegionAngle}, \end{aligned} \tag{5.17}$$

with

$$\begin{gathered} \mathbf{d}_a = \begin{pmatrix} \cos\theta_a \\ \sin\theta_a \end{pmatrix}, \quad \mathbf{d}_b = \begin{pmatrix} \cos\theta_b \\ \sin\theta_b \end{pmatrix}, \quad \text{and} \\ \tilde{\mathbf{w}}_a = \mathbf{w}_a - \texttt{d\_EndPtShift}\, \mathbf{d}_a, \quad \tilde{\mathbf{w}}_b = \mathbf{w}_b - \texttt{d\_EndPtShift}\, \mathbf{d}_b. \end{gathered} \tag{5.18}$$

This condition says that the distance between the two end-points must be closer than `d_MaxInterEndptDistance`, and the end-point direction vectors should point toward each other. This is illustrated in Figure 5.12. If the distance between the end-points is really small, the criterion should be less sensitive to direction. This is achieved by shifting the corners of the wedge-shaped regions over a distance `d_EndPtShift`, denoted by $\tilde{\mathbf{w}}_a$ and $\tilde{\mathbf{w}}_b$.

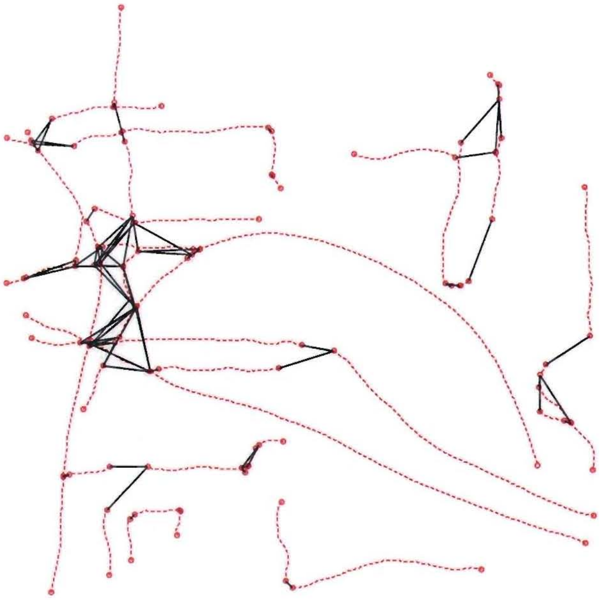


Figure 5.13: *Example of a path graph. The red dashed line indicate the extracted paths, and the black lines indicates connections between the end-points of the paths.*

We assign to each connection in the path graph a certainty measure, which includes certainty of both paths, distance, and end-point directions. We choose for this certainty measure

$$s_{\text{connect}}(\mathbf{w}_a, \mathbf{d}_a, s_a, \mathbf{w}_b, \mathbf{d}_b, s_b) = s_a\, \mathbf{d}_a \cdot \frac{(\mathbf{w}_b - \mathbf{w}_a)}{||\mathbf{w}_b - \mathbf{w}_a||} + s_b\, \mathbf{d}_b \cdot \frac{(\mathbf{w}_a - \mathbf{w}_b)}{||\mathbf{w}_a - \mathbf{w}_b||}, \tag{5.19}$$

where $\mathbf{w}_a$ and $\mathbf{w}_b$ are the end-point coordinates, $\mathbf{d}_a$ and $\mathbf{d}_b$ the unit end-point direction vectors, and s_a and s_b the saliency of the two paths (defined in Equation 5.3).

The algorithm to create the graph simply checks the condition in Equation 5.17 for all end-points of all paths, and creates connections if the condition holds.

Algorithm 4: *Create path graph connections*

```
For i=1 to $|\mathcal{Q}|$
  For j=i+1 to $|\mathcal{Q}|$
    If (Equation 5.17 holds for an end-point of $\mathcal{P}_i$ together with an end-point of $\mathcal{P}_j$)
    then Create connection $\{\mathcal{P}_i, \mathcal{P}_j\}$ in $\mathcal{G}$, and
      calculate connection certainty according to Equation 5.19.
```

Figure 5.13 gives an example.

5.3 Electrode extraction and grouping

This section describes the detection of electrodes on EP catheters utilizing the spacing between the electrodes.

5.3.1 Extraction of electrode candidates

In this step, a set of electrode candidates is extracted from the blobness image B. The resulting set of candidates is denoted by

$$\mathcal{B} = \{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_{|\mathcal{B}|}\}, \tag{5.20}$$

where $\mathbf{b}_i$ is the coordinate of the i-th candidate. The certainty of the blob at $\mathbf{b}_i$ is $B(\mathbf{b}_i)$. So the certainty is equal to the response strength in the blobness image. The algorithm for electrode candidate extraction is as follows.

Algorithm 5: *Extraction of most salient electrode candidates*

- In blobness image B, only keep the pixels that are part of one of the extracted paths in $\mathcal{Q}$. Extract the local maxima in B. A pixel is considered a local maximum if its value is larger than all the surrounding pixels within a certain distance `d_LocalMaxExtract`.
- Keep the `n_BlobExtract` highest extracted local maxima as electrode candidates in set $\mathcal{B}$.
- For every electrode candidate, check, on which path in $\mathcal{Q}$ it is situated. Store this information for the next steps.

5.3.2 Creation of electrode graphs

The next step is to create a graph with the electrode candidates $\mathcal{B}$ as nodes. Similar to the path graph (Subsection 5.2.6), non-directional connections are created between all nodes that could be adjacent electrodes on an EP catheter. These connections are the *edges* of the graph, denoted by $\mathcal{E}$.

For the next step to be effective, the number of abusive connections between nodes should be as small as possible. This means we need to define a strict set of rules, specifying whether a connection should be made or not. Nevertheless, it is safer to make more connections than necessary instead of the other way around.

To create the graph connections, we use the following commutative conditions:

- *Electrode distance.* We allow a minimum and maximum Euclidean pixel distance between two adjacent electrodes $\mathbf{b}_i$ and $\mathbf{b}_j$ on a catheter

$$\texttt{d\_MinInterElectrode} \leq ||\mathbf{b}_i - \mathbf{b}_j|| \leq \texttt{d\_MaxInterElectrode}. \tag{5.21}$$

  `d_MaxInterElectrode` must be chosen such that the inter-electrode distance in the image is never larger. This is determined by the maximum physical distance of adjacent electrodes on the real catheter, related to the physical dimensions of a pixel in the X-ray image, which is dependent on the settings of the X-ray equipment. Due to varying distances between electrodes on different catheters and because of projection, the inter-electrode distance can be smaller or even zero. `d_MinInterElectrode` is therefore determined by the minimum distance, for which the blob detection will still be able tell two blobs apart.

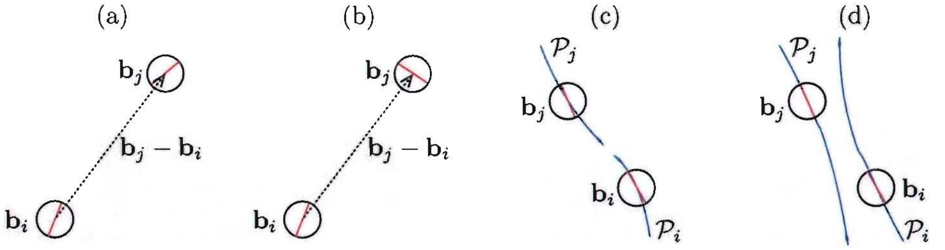


Figure 5.14: *Some examples on electrode graph creation, demonstrating the orientation criterion and path end-point criterion. The circles indicate the electrode candidate positions, the red lines inside the circles indicates the orientations $\Psi(\mathbf{b}_i)$ resp $\Psi(\mathbf{b}_j)$. (a) The angle between the connecting vector $\mathbf{b}_j - \mathbf{b}_i$ and orientations $\Psi(\mathbf{b}_i)$ and $\Psi(\mathbf{b}_j)$ is small, so the electrodes are connected. (b) The angle $\Psi(\mathbf{b}_j)$ is wrong, meaning that candidate $\mathbf{b}_j$ will belong to another structure. No connection is created. (c) Both candidates belong to different paths (indicated by the blue lines). However, the orientations suggest that a connection should be made, and so do the end-points of both paths. (d) The candidates are oriented in exactly the same way as in (c), but the positions of the end-points of the paths suggest that the electrodes should not be connected.*

- *Electrode orientation.* We know that the catheter has a certain stiffness. Therefore, we only allow a maximum difference in angle between the orientation in orientation map Ψ at candidates $\mathbf{b}_i$ and $\mathbf{b}_j$, and the vector $\mathrm{b_i} - \mathrm{b_j}$ connecting these candidates. This is illustrated in Figure 5.14a-b. It leads to the following criterion

$$\begin{aligned} &\left| \frac{\mathbf{b}_i - \mathbf{b}_j}{||\mathbf{b}_i - \mathbf{b}_j||} \cdot \begin{pmatrix} \cos \Psi(\mathbf{b}_i) \\ \sin \Psi(\mathbf{b}_i) \end{pmatrix} \right| \geq \texttt{ca\_MaxElectrAngleChange} \\ \wedge &\left| \frac{\mathbf{b}_i - \mathbf{b}_j}{||\mathbf{b}_i - \mathbf{b}_j||} \cdot \begin{pmatrix} \cos \Psi(\mathbf{b}_j) \\ \sin \Psi(\mathbf{b}_j) \end{pmatrix} \right| \geq \texttt{ca\_MaxElectrAngleChange}, \end{aligned} \tag{5.22}$$

  where `ca_MaxElectrAngleChange` is the cosine of the maximal allowable angle.

- *Knowledge on paths.* If two candidates belong to the same path, we do not need to check the orientation condition above anymore, because we already imposed a curvature constraint on the paths. If two candidates belong to different paths, however, the orientation condition should be checked and it also makes sense to check whether the two paths have a connection in the path graph (see Figure 5.14c-d), i.e.

$$\{\mathcal{P}_i, \mathcal{P}_j\} \in \mathcal{G}, \tag{5.23}$$

  where $\mathcal{P}_i$ is the path to which $\mathbf{b}_i$ belongs, $\mathcal{P}_j$ is the path to which $\mathbf{b}_j$ belongs, and $\mathcal{G}$ is the set of connections between paths (Subsection 5.2.6). The check for a connection in the path graph is not sufficient. It can happen that the paths have a long trajectory between the two electrodes instead of a short and approximately straight connection. A possible way to overcome this is by checking the sum of distances of the electrode candidates relative to the closest end-point of the paths they belong to. This distance should be smaller than `d_MaxInterElectrode`.

$$\min_{k=1,2} ||\mathbf{b}_i - \mathbf{w}_k(\mathcal{P}_i)|| + \min_{k=1,2} ||\mathbf{b}_j - \mathbf{w}_k(\mathcal{P}_j)|| \leq \texttt{d\_MaxInterElectrode}, \tag{5.24}$$

  where $\mathbf{w}_k(\mathcal{P}_i)$ is the coordinate of the first ($k = 1$) resp second ($k = 2$) end-point of path $\mathcal{P}_i$.

These conditions brings us to the following graph connection algorithm:

Algorithm 6: *Create electrode graph connections*

```
For i=1 to $|\mathcal{B}|$
  For j=i+1 to $|\mathcal{B}|$
    If (d_MinInterElectrode $\leq ||\mathbf{b}_i - \mathbf{b}_j|| \leq$ d_MaxInterElectrode)
      and ($\mathbf{b}_i$ and $\mathbf{b}_j$ belong to the same path
           or conditions in Equation 5.22, 5.23, and 5.24 are all fulfilled)
    then Create dual-sided connection $\{\mathbf{b}_i, \mathbf{b}_j\}$ in $\mathcal{E}$.
```

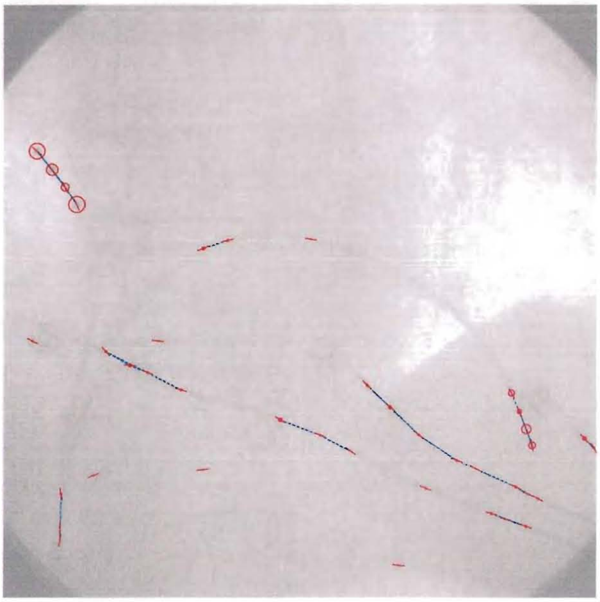


Figure 5.15: *Example of electrode graph. 40 candidates were extracted from the context enhanced blobness image B. The red circles indicate the electrode candidates, the radius shows the certainty. The red lines show the orientations at the positions of the electrode candidates. The blue dashed lines show the connections.*

Figure 5.15 shows an example.

5.3.3 Electrode group detection

In this step, a number of most promising *electrode group candidates* are extracted. An electrode group candidate is a list of electrode coordinates $\mathcal{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_{|\mathcal{C}|}\}$ that represent the electrodes situated on a single catheter (if the extraction is correct). An electrode group is a subset of $\mathcal{B}$, $\mathcal{C} \subset \mathcal{B}$, forming a *path* in the electrode graph, i.e. $\{\mathbf{c}_i, \mathbf{c}_{i+1}\} \in \mathcal{E}$, $\forall\, 1 \leq i \leq |\mathcal{C}| - 1$.

The electrodes in the list are ordered as they appear on the catheter itself. The certainty of an electrode group is defined as

$$s(\mathcal{C}) = \sum_{i=1}^{|\mathcal{C}|} B(\mathbf{c}_i). \tag{5.25}$$

The method presented here is designed such that it is not needed to exactly know the properties of the EP catheters in use (i.e. number of electrodes, distances between electrodes), but the algorithm is expected to be more robust if more specific knowledge is supplied.

We only used conditions involving 2 candidates for creating the electrode graph. Now, we use conditions describing contextual relations between more than 2 adjacent electrode candidates. The following additional conditions are used:

- *Catheter stiffness.* We check for the angle between the vector connection candidate $\mathbf{b}_i$ and $\mathbf{b}_j$, and $\mathbf{b}_j$ and $\mathbf{b}_k$, i.e.

$$\frac{(\mathbf{b}_i - \mathbf{b}_j) \cdot (\mathbf{b}_j - \mathbf{b}_k)}{||\mathbf{b}_i - \mathbf{b}_j||\, ||\mathbf{b}_j - \mathbf{b}_k||} \leq \texttt{ca\_MaxElectrGroupAngleChange}. \tag{5.26}$$

- *Number of electrodes.* We restrict the allowable number of electrodes in an electrode group.

$$|\mathcal{C}| \leq \texttt{n\_MaxElectrodes}. \tag{5.27}$$

If we know the number of electrodes on the catheter in use, `n_MaxElectrodes` should be set to this value. Otherwise an upper bound should be specified. We also define a minimum number of electrodes for a valid group

$$|C| \geq \texttt{n\_MinElectrodes}. \tag{5.28}$$

- *Alternating distance between electrodes.* On most EP catheters, electrodes appear in pairs (see the examples in Figure 2.3, page 15). This means a pair of electrodes have a small distance, and a larger distance to the other adjacent electrodes. Furthermore, we do not know the exact distance between electrodes in the image, due to the projection from 3D to 2D, but we do know that due to the stiffness of the catheter, the *relative change in distance* between the different electrodes of one catheter can not be large. This gives the following alternating distance criterion. Suppose we want to check if an electrode candidate $\mathbf{b}$ is a valid extension at the back side of electrode group candidate $C = \{\mathbf{c}_1, \ldots, \mathbf{c}_{|C|}\}$ with $|C| \geq 3$), then the following condition should hold

$$\frac{\left| ||\mathbf{b} - \mathbf{c}_{|C|}|| - d_2 \right|}{\max(d_1, d_2)} \leq \texttt{r\_MaxElectrRelDistChange}, \tag{5.29}$$

where $d_1 = ||\mathbf{c}_{|C|-1} - \mathbf{c}_{|C|}||$, and $d_2 = ||\mathbf{c}_{|C|-2} - \mathbf{c}_{|C|-1}||$. This is illustrated in Figure 5.16. The advantage of this measure is that we do not need to know in advance the ratio of the distance between electrodes forming a pair and the distance between electrodes of adjacent pairs, which is EP catheter dependent. If we do know this, however, we can include this in the criterion. This will lead to a criterion that is more discriminatory.

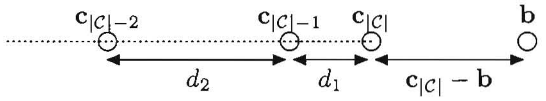


Figure 5.16: *Illustration of the alternating distance criterion. Electrode candidate* $\mathbf{b}$ *is a valid addition to electrode group* C *if the distance between* $\mathbf{b}$ *and* $\mathbf{c}_{|C|}$ *is approximately similar to distance* d_2.

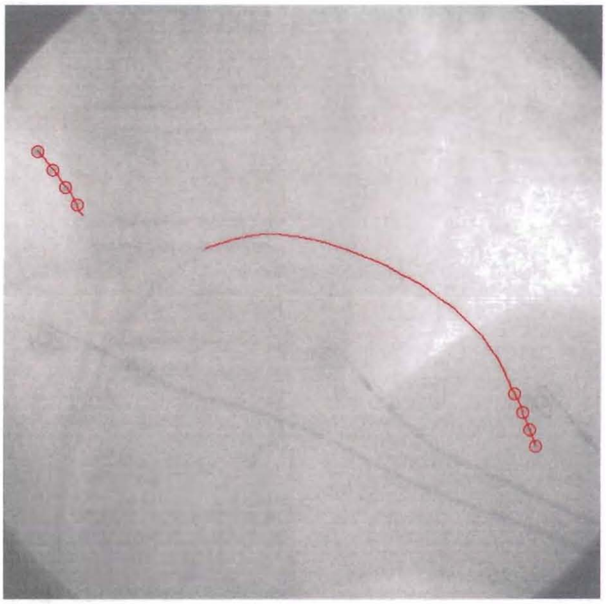


Figure 5.17: *Extracted catheter tips, using the graph of Figure 5.15. The electrode groups were detected as described in Subsection 5.3.3. Afterwards, the corresponding tip paths were extracted using Subsection 5.3.4.*

- *Threshold on electrode certainty.* If the distance criterion does not hold, then still it might be appropriate to add an electrode. One electrode on the catheter could be missed, leading to a disturbed regular alternating distance pattern. An electrode candidate $\mathbf{b}$ that does not fulfill the distance criterion for a group $\mathcal{C}$, can still be a valid addition to the group if its certainty is larger than the threshold `r_MinRelativeElectrCertainty`. This threshold is defined relative to the average certainty of the electrodes already in $\mathcal{C}$, i.e.

$$B(\mathbf{b}) \geq \texttt{r\_MinRelativeElectrCertainty}\,\frac{s(\mathcal{C})}{|\mathcal{C}|}. \tag{5.30}$$

The electrode grouping algorithm is as follows:

Algorithm 7: *Electrode grouping*

```
• Each electrode candidate (graph node) has a mark, indicating whether the node is
  already visited. All marks are initiated to zero (unmarked).
• While there are unmarked nodes:
    – Select the unmarked node $\mathbf{b}$ with highest certainty $B(\mathbf{b})$.
    – Create electrode groups consisting of 3 electrodes for all possible combinations of
      the current electrode with its neighbors. These are groups that fulfill the stiffness
      criterion of Equation 5.26. All formed groups are enqueued, in a queue where all
      unfinished electrode groups are stored, waiting for further extension. All visited
      nodes are marked.
    – While there are enqueued groups:
        * Get enqueued group, say $\mathcal{C}$.
        * Do
            · Collect all electrode neighbors of $\mathbf{c}_{|\mathcal{C}|}$ (i.e. the last electrode in $\mathcal{C}$) that
              fulfill the stiffness criterion of Equation 5.26. Mark all visited nodes.
            · If there is a neighbor $\mathbf{b}$ fulfilling the distance criterion of Equation 5.29
              then extend electrode group $\mathcal{C}$ with that electrode. If more neighbors
              fulfill Equation 5.29, select the best one, i.e. the one with the smallest
              value in Equation 5.29.
            · Otherwise, select the neighbor $\mathbf{b}$ with highest $B(\mathbf{b})$. If $B(\mathbf{b})$ fulfills the
              electrode certainty threshold criterion of Equation 5.30 then add $\mathbf{b}$ to $\mathcal{C}$
            · If $\mathcal{C}$ can not be extended anymore to this site, and the other side can
              still be extended, reverse $\mathcal{C}$ to try the other side.
          While $\mathcal{C}$ can be extended.
        * The current group is now finished. If the number of electrodes in the current
          group (i.e. $|\mathcal{C}|$) is smaller than n_MinElectrodes, discard $\mathcal{C}_i$.
        * If the number of electrodes in the current group (i.e. $|\mathcal{C}|$) is larger than
          n_MaxElectrodes:
          Repeat
            · If $s(\mathbf{c}_1) < s(\mathbf{c}_{|\mathcal{C}|})$ then remove $\mathbf{c}_1$, otherwise remove $\mathbf{c}_{|\mathcal{C}|}$. Remove the
              electrode at the end of the group with lowest certainty.
          Until $|\mathcal{C}| =$ n_MaxElectrodes.
  The result is a set of electrode groups. All nodes contained in the groups are marked.
• Sort the set of extracted group candidates with respect to their certainty (defined in
  Equation 5.25). This results in a list $\{\mathcal{C}_1, \ldots, \mathcal{C}_n\}$ fulfilling $s(\mathcal{C}_i) \geq s(\mathcal{C}_{i+1})\ \forall\, 1 \leq i < n$.
• Iterate through sorted group candidate list:
    – If all electrodes in the group are marked: return this group, and unmark all
      nodes contained in the group, otherwise discard the group.
    – Continue until n_ElectrGroupExtract of electrode groups are returned.
```

In the end, we get a set of `n_ElectrGroupExtract` most salient electrode groups $\{\mathcal{C}_1, \ldots, \mathcal{C}_k\}$ with $k =$ `n_ElectrGroupExtract`, and $\mathcal{C}_1 \cap \mathcal{C}_2 \cap \ldots \cap \mathcal{C}_k = \emptyset$. Figure 5.17 shows an example.

5.3.4 Creation of catheter tip paths

From a group of electrodes $\mathcal{C}$, consisting of a list of coordinates indicating the electrode positions, we want to obtain the path connecting the electrodes. We call such a path a *catheter tip path* $\mathcal{P}_{\text{tip}}$. Here, with *tip* the part of the catheter with electrodes is meant. We also want to decide which side of the tip path is the real end-point of the catheter, and which side should be extended. For creating the tip path we use the extracted paths from Section 5.2.

To decide which side to extend, the idea is to first try to extend the tip path to both sides for a predefined length `n_PixelsDirectionDecision`. The extension path at the side of the tip that must be extended, will most likely exhibit higher ridgeness values than the other side. We decide on this by taking the minimum along both extension paths and selecting the direction with highest minimum value as extension side.

Figure 5.17 shows an example of extracted catheter tip paths. The algorithm is as follows

Algorithm 8: *Create catheter tip $\mathcal{P}_{tip}$ path from electrode coordinates $\mathcal{C}$*

```
• Create inter-electrode path:
  Init $\mathcal{P}_{\text{tip}} = \emptyset$;
  For i=1 to $|\mathcal{C}| - 1$
    If($\mathbf{c}_i$ and $\mathbf{c}_{i+1}$ belong to the same path $\mathcal{P}$)
      Then add the part of path $\mathcal{P}$ interconnecting $\mathbf{c}_i$ and $\mathbf{c}_{i+1}$ to $\mathcal{P}_{\text{tip}}$
      Else interpolate between the end-point of the paths connecting the two electrodes
• Decide which side of the path $\mathcal{P}_{\text{tip}}$ should be extended:
    – Starting from both electrodes forming the end-points of $\mathcal{P}_{\text{tip}}$, create extension
      paths $\mathcal{P}_{\text{ext,L}}$ and $\mathcal{P}_{\text{ext,R}}$ of length n_PixelsDirectionDecision, by concatenating
      paths that have the best connection certainty according to Equation 5.19, until
      the extension paths have length n_PixelsDirectionDecision (if possible).
    – Determine minimum value of ridgeness $R$ along the pixels in path $\mathcal{P}_{\text{ext,L}}$ and
      $\mathcal{P}_{\text{ext,R}}$. The side with highest minimum pixel value is chosen as the side to
      extend.
```

The interpolation between the end-points $\mathbf{w}_a$ and $\mathbf{w}_b$ of two different paths is achieved by simply creating a straight line in between, i.e.

$$\mathbf{l}(t) = \mathbf{w}_a + t(\mathbf{w}_b - \mathbf{w}_a). \tag{5.31}$$

This straight line is sampled and rounded, such that the path requirements in Subsection 5.2.1 hold. The straight line interpolation in practice appears to be a sufficient approximation.

5.4 Path grouping

The path grouping algorithm uses the catheter tip paths as *seed* for further extension, in order to detect a longer part or the entire catheter. This is achieved by combining *paths* into longer paths called *super paths.* The problem is to a large extent similar to the electrode grouping problem and so is our approach.

5.4.1 Creation of possible extensions for catheter tips

Given a seed path $\mathcal{P}_{\text{tip}}$, we want to extract all plausible extensions, by "walking" through the created path graph. The reason to first extract all extensions that could be correct, instead of searching for a shortest path in the graph using a cost function on the edges, is because such a cost function is still based on *relatively local criteria*, and not really on *global criteria* on the structure of interest.

The method searches for possible trajectories through the created path graph by extending the seed path. The resulting candidates are called *super paths.* A super path $\mathcal{S}$ is defined as a list of (possibly reversed) paths. For example, we can have a super path $\mathcal{S} = \{\mathcal{P}_a, \breve{\mathcal{P}}_b, \breve{\mathcal{P}}_c, \mathcal{P}_d\}$. The ordering of the paths in the super path is such, that the paths forming the super path are adjacent paths in the path graph and the first one is the seed path.

For the search process we need a *stopping criterion* to make sure that the extension stops at an appropriate location. To also restrict the number of extracted candidates, a *splitting criterion* is used. If there is more than one extensions possible, this criterion decides if the second-best (third-best, etc.) extensions are certain enough relative to the best extension, up to a certain maximum number of splits.

There are three reasons to stop:

- We arrived at a path with an end-point that has no connections with other paths.
- The end-point comes closer than `d_BorderDistanceStop` to the border of the image.

  $$d_{\text{border}}(\mathbf{w}(\mathcal{S})) < \texttt{d\_BorderDistanceStop}, \tag{5.32}$$

  where $\mathbf{w}(\mathcal{S})$ is the current end-point of the super path, and $d_{\text{border}}(\mathbf{w})$ is the distance to the border of position $\mathbf{w}$ in the image, defined as the Euclidean distance to the closest image border point.
- The super path has reached a certain maximum length `n_MaxSuperPathLength`.

  $$\text{length}(\mathcal{S}) > \texttt{n\_MaxSuperPathLength}, \tag{5.33}$$

  where length($\mathcal{S}$) is the sum of number of pixels of all paths it consists of

  $$\text{length}(\mathcal{S}) = \sum_{i=1}^{|\mathcal{S}|} |\mathcal{P}_i|. \tag{5.34}$$

For the splitting criterion, we use the connection certainty measure of Equation 5.19. Suppose we have k different possible extensions for the current super path, with certainty values denoted by $s_{\text{connect}}[i]$, sorted by certainty such that $s_{\text{connect}}[1]$ is largest. Then the splitting criterion, specifying whether the extension with index k should be made, is

$$\frac{s_{\text{connect}}[1] - s_{\text{connect}}[k]}{s_{\text{connect}}[1]} \leq \texttt{r\_MaxCertaintyChange} \quad \wedge \quad k \leq \texttt{n\_MaxNumberOfSplits}, \tag{5.35}$$

where `r_MaxCertaintyChange` is the maximum allowable difference in certainty compared to the best extension, and `n_MaxNumberOfSplits` is the maximum number of splits to make.

Algorithm 9: *Generate catheter candidates, starting from seed $\mathcal{P}_{tip}$*

```
• $\mathcal{S} = \{\mathcal{P}_{\text{tip}}\}$
• Put $\mathcal{S}$ in queue
• While there are super paths in the queue:
    – Pop a super path $\mathcal{S}$ from queue
    – While (Stopping criterions of Equation 5.32 and Equation 5.33 do not hold and
      there are possible extensions):
        * If there is more than one possible extension, fulfilling Equation 5.35
          Then for all valid extensions except for the best one: create a copy of $\mathcal{S}$,
          add the extension, and put in the queue.
        * Take best possible extension path and add it to $\mathcal{S}$.
    – Return current super path $\mathcal{S}$ as catheter candidate.
```

The result is a set $\mathcal{W}$ of super path candidates, for every seed path. Figure 5.18 shows an example. In this example, six extension candidates are shown for both catheter tip seeds.

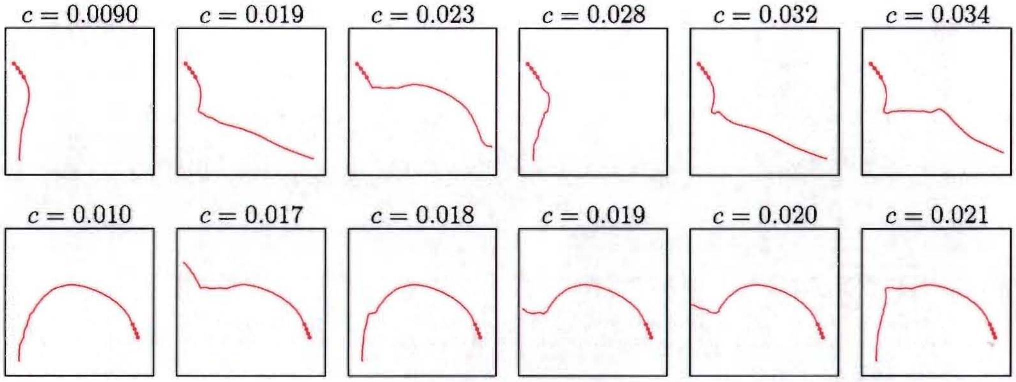


Figure 5.18: *Illustration of the 6 best extensions (according to the criterion in Equation 5.38) for the two extracted catheter tip paths (Figure 5.17). The indicated values for c are the result of the calculation Equation 5.38 on the candidates. For both catheters, the extension candidate with the lowest value for c indeed represent the correct catheter.*

5.4.2 Selecting the best extension using a global criterion

To select the path that is most likely to be our catheter, we use the knowledge that

- low curvature is favored over larger curvature;
- a low *change in curvature* is favored over larger change in curvature.

In Subsection 5.2.4, we estimated the curvature $\rho[i]$ at position i on a path (Equation 5.12). Here, the same curvature estimation is used, but now we take the average of the absolute value of the curvature of the entire path

$$\overline{\rho}(\mathcal{P}) = \frac{1}{|\mathcal{P}|-2} \sum_{i=2}^{|\mathcal{P}|-1} |\rho[i]| \,. \tag{5.36}$$

For the change in curvature we use the standard deviation of the absolute curvature

$$\sigma_\rho(\mathcal{P}) = \sqrt{\frac{1}{|\mathcal{P}|-2} \sum_{i=2}^{|\mathcal{P}|-1} (|\rho[i]| - \overline{\rho})^2}. \tag{5.37}$$

The product of these two measures are used as the global cost function

$$c_{\text{global}}(\mathcal{P}) = \overline{\rho}(\mathcal{P}) \cdot \sigma_\rho(\mathcal{P}). \tag{5.38}$$

The algorithm is:

Algorithm 10: *Select best super path candidate in set $\mathcal{W}$*
For all extension candidates $\mathcal{S}$ in $\mathcal{W}$:

- Convert super path $\mathcal{S}$ to a normal path representation, by concatenating all paths, and interpolating between end-points with straight lines.
- Calculate global cost function of Equation 5.38.

Select the candidate with minimum value for c_{global}.

In Figure 5.18 the values for the measure c_{global} are shown for the different candidates. In Figure 5.19, the resulting best catheters are superimposed on the original image.

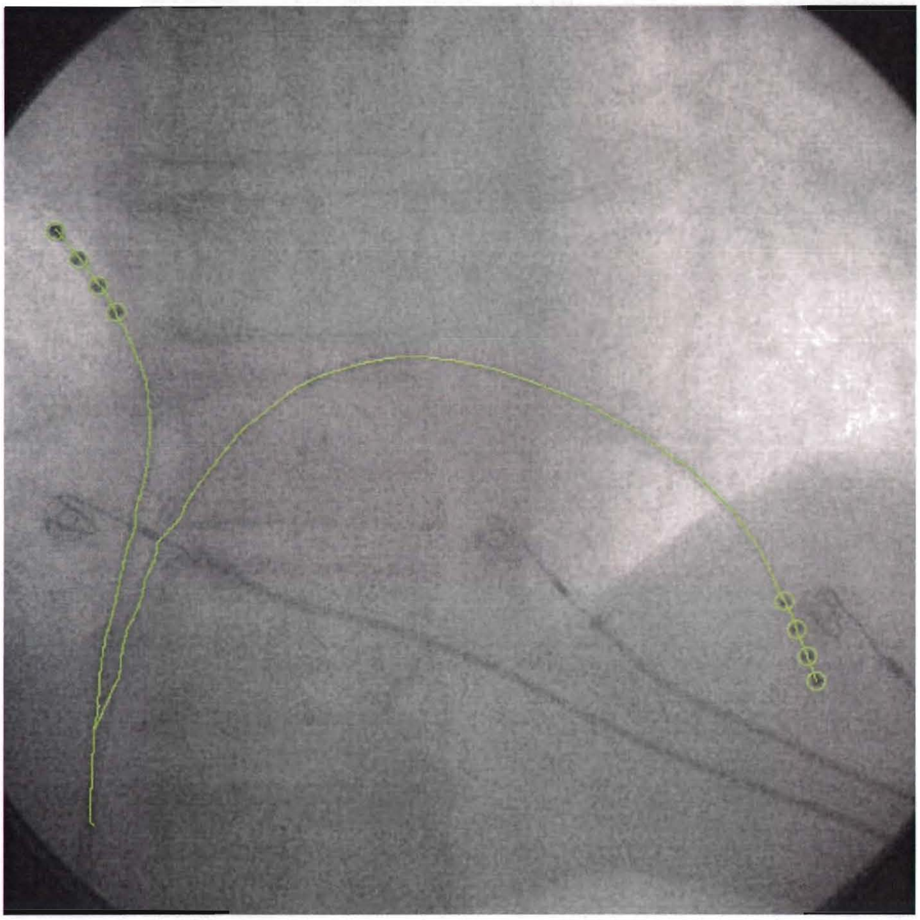


Figure 5.19: *Final extracted catheters in the example image.*

5.5 Algorithm parameters

In Table 5.2 all parameters are summarized. There are quite a lot of parameters in the EP catheter extraction algorithm. However, we expect that most parameters are not decisive. The effect of most parameters is that the number of possible solutions is restricted, so if we set these parameters to permissive values, the effect of each individual parameter will be small. The column "Value" shows values for the parameters that are optimized as will be described in Subsection 6.2.2. The column "Range" gives a indication for reasonable ranges for each parameter.

5.6 Conclusions

In this chapter, we derived an algorithm to extract the EP catheters. The context enhanced feature images of Chapter 4 are used to extract paths and electrode candidates. We use the certainty of the features in the feature images, orientation information, and spatial information to make decisions. The strategy is to group the paths and electrode candidates to the final EP catheters in multiple steps, limiting the number of possibilities in every step. For this purpose, all kinds of conditions are derived from knowledge we have about EP catheters. In the end, a global measure is used to pick the final, best candidate. During this process a lot of parameters are introduced, whose individual values should have as little impact as possible. The robustness of the method still has to be proven. In Chapter 6 this will be investigated more thoroughly.

We described methods that in principle can extract the entire catheter. In clinical practice, however, it is probably sufficient to only extract the last part (for instance, the part of the catheter with electrodes plus an additional part of the same length). This does not lead to major changes in the required algorithm. If one is really only interested in the tip, one can stop after the creation of the catheter tip paths (Subsection 5.3.4). If one wants a somewhat larger part of the catheter, one can extend it with the next steps and choose a small value for parameter `n_MaxSuperPathLength`.

The number of possible EP catheter configurations in images is almost endless. It is hard to take all situations into account. The following situations can be hard to detect, because of ambiguities that may lead to failures:

- If the electrodes are not visible at all, due to projection or too low dose. This is problematic because the found electrode groups function as seeds. To solve this dilemma, a fall-back seed selection mechanism could be included. For instance, a path with highest certainty could be used as a seed. Such a seed selection mechanism makes the path grouping method also suitable for other detection problems with line structures, such as guide wire detection.
- If two catheters fall together in an image. Especially if their tips fall together.
- If a catheter shows a large gap, because background structures cause too much X-ray radiation to be absorbed. Small gaps will be closed by the tensor voting step and the path grouping step, but for larger gaps this becomes more complicated.
- If a catheter shows a kink.
- If other line structures appear in the image, for instance ECG stickers.
- If two line structure cross each other at a small angle.

It is impossible to solve these problems entirely in the *spatial* domain. So it is essential to use the *temporal* domain as well. An ambiguity in a single frame can be resolved using the previous frames. Both temporal and spatial information should be used simultaneously. The scope of this project does not include the use of temporal information, but it is advisable to investigate this in the future.

Path extraction (Section 5.2)				
Name	*Value*	*Range*	*Description*	*Subsec.*
n_PruningSteps	1	0 – 8	Number of pruning steps on thinned image	5.2.2
n_PathExtract	40	20 – 150	Number of most salient paths to extract	5.2.3
th_MinPathLength	8	0 – 12	Minimal length of extracted path	
α_d	0.85	0.7 – 0.92	Decay of IIR filter for direction estimation	5.2.4
ca_CurvatureSplit	0.82	0.7 – 0.95	Curvature threshold for splitting a path	
d_PathEndptSplit	4	2 – 10	Max. distance of a path to end-point region of other path, in order to split the path	5.2.5
d_PathEndptAnisotropy	8	0 – 20	Anisotropy size of end-point region of a path	
d_MaxInterEndptDistance	60	10 – 100	Max. Euclidean distance between two endpoints	5.2.6
ca_PathConnectRegionAngle	0.5	0–0.7	Cosine of maximum angle between two endpoints	
d_EndPtShift	5	0–10	Shift distance of wedge-shaped regions that infers graph connections	

Electrode extraction and grouping (Section 5.3)				
Name	*Value*	*Range*	*Description*	*Subsec.*
d_LocalMaxExtract	4	1 – 6	Size of neighborhood for local max. extraction	5.3.1
n_BlobExtract	40	20 – 150	Number of electrode candidates to extract	
d_MinInterElectrode	6	3 – 10	Minimum distance between adjacent electrodes	5.3.2
d_MaxInterElectrode	45 [b]	30 – 60	Maximum distance between adjacent electrodes	
ca_MaxElectrAngleChange	0.7	0.6 – 0.9	Min. cosine of angle between electrodes for creating graph connections	
n_ElectrGroupExtract	[a]	1 – 6	Number of best electrode groups to extract	5.3.3
n_MaxElectrodes	20 [b]	4 – 40	Number of electrodes on catheters in use, or upper bound of number of electrodes in a group (if unknown catheter type)	
n_MinElectrodes	3	3 – 10	Min. allowable number of electrodes a in group	
r_MaxElectrRelDistChange	0.25	0.1 – 0.5	Max. allowable relative deviation in distance	
ca_MaxElectrGroupAngleChange	0.9	0.6 – 0.95	Min. cosine of angle between electrodes for creating electrode groups	
r_MinRelativeElectrCertainty	0.4	0.2 – 0.6	Min. relative certainty for electrode to be grouped	
n_PixelsDirectionDecision	40	10 – 60	Number of pixels that should be taken into account to decide which direction to extend from a catheter tip	5.3.4

Path grouping (Section 5.4)				
Name	*Value*	*Range*	*Description*	*Subsec.*
d_BorderDistanceStop	50	20 – 100	Max. distance to border for stopping criterion to be true	5.4.1
n_MaxSuperPathLength	700	300 – 1000	Max. allowable length of a super path	
r_MaxCertaintyChange	0.4	0.2 – 0.8	Max. relative difference in certainty between best extension candidate and other extension candidates	
n_MaxNumberOfSplits	3	1 – 5	Max. number of edges to follow starting from a path in the path graph	

[a] This value depends on the number of catheters that are currently in use.

[b] The optimal value depends on the type of catheters that are currently in use.

Table 5.2: *Overview of EP catheter extraction algorithm parameters. The values in column "Value" are optimized as described in Subsection 6.2.2. The column "Range" specifies ranges for the parameters. These ranges are only an indication for reasonable values and do not imply that values outside this will not work (Table 5.1 shows the domains, in which the different parameter classes should be defined). The parameters concerning distance (starting with "d_") are expressed in pixels for an image of* 512×512 *pixels.*

Chapter 6

Evaluation

In the previous three chapters, we introduced a complete method to detect EP catheters using spatial information. The purpose of this chapter is to evaluate how well the method works.

In principle, only the end result (the extracted EP catheters) really counts. However, evaluation of intermediate results offers more insight in what works and what goes wrong. It also makes it easier to tune parameters, because every step can be tuned separately.

We consider the generated feature images (Chapter 3 and Chapter 4) as important intermediate results. Therefore, we in Section 6.1 will evaluate the quality of the ridgeness feature images. The focus is on robustness to noise, difference between different combinations of filters, and on optimal settings of the parameters. Then, in Section 6.2 we will evaluate the real end result of the EP catheter extraction algorithm. Again, robustness to noise is an important issue. Finally, in Section 6.3, we will evaluate the computational aspects.

A lot of literature exists describing methods to evaluate medical image analysis techniques, see e.g. [35, Chapter 10]. A real thorough evaluation, as needed before a method can be introduced in clinical practice, is outside the scope of this project. This is not possible due to time limitations, known situations that the algorithm can not yet handle, and because the number of available images is too limited. We also have no other algorithms for EP catheter detection to compare the method to. Within Philips Medical Systems no other EP catheter detection algorithms were designed, and hardly any literature was found on EP catheter detection (only in [27], but that work is not comparable, because it only describes a method to detect EP catheter electrodes and does not extract an EP catheter). Nevertheless, a comparison can be made between different variants. For instance, the TV (tensor voting) step can be skipped to show the added value of this step.

To keep this chapter orderly, some detailed tables and figures that are not necessary to get the essence of the results, are put in Appendix A.

6.1 Evaluation and tuning of feature image quality

The goal of this section is to evaluate the quality of the feature images obtained for different combinations of filters. The questions we try to answer are the following:

- Which local feature detector performs best?
- Is there a clear added value of TV, i.e. is the context enhanced ridgeness image R considerably better than the local ridgeness r?
- What is the robustness to noise, for the different feature images?
- What parameter settings are optimal?

The general strategy is as follows:

- The ground truth is segmented for a set of test images.

- Evaluation measures are defined with the goal to express the quality of a feature image in a number.
- For each test image, different feature images are generated. On these feature images, the evaluation measures are calculated to see which combination of filters and settings is best (according to the evaluation measures).

For every separate step (local feature detection, TV) we will try to find optimal parameter values *separately*, under the *assumption* that the optimal parameter values according to the output of a single step, is also the optimal parameter value in the entire EP catheter extraction algorithm. Otherwise, the number of parameters to optimize in one step would be too large.

6.1.1 Test images and ground truth

Figure 6.1 shows the EP catheter images used in this test. EP3, EP4, EP5, and EP6 are *clinical images*, i.e. images recorded in clinical practice. EP1 and EP2 are *phantom images*, i.e. images recorded in a non-clinical situation with the use of a thorax phantom to imitate a realistic situation. We acquired these phantom images at different dose levels, see Figure A.1 and Figure A.2 (Appendix A). To get different dose levels, the beam current I_{beam} (unit mAs, milli-ampère-second) was varied, while the voltage of the X-ray beam was fixed. The relation between I_{beam} and the signal-to-noise ratio (SNR) of the resulting image is as follows. I_{beam} is proportional to the expected number of radiated X-ray photons, i.e. $I_{\text{beam}} \sim E(\text{photons})$ over a certain time span. The radiation of the photons over time is a poisson process, implying that $E(\text{photons}) \sim \text{var}(\text{photons})$. Assuming that the poisson distribution of the photons is the most important source of noise, the SNR of the resulting X-ray image is equal to $\frac{E(\text{photons})}{\sqrt{\text{var}(\text{photons})}}$, so SNR $\sim \sqrt{I_{\text{beam}}}$. The indicated current values are thus an indication for image quality[1].

We segmented the ground truth of the 6 images manually. The pixels in the image are divided in 3 classes:

- The *foreground pixels*: the pixels that belong to the *centerline of the EP catheter*. The values of these pixels are denoted by the set $\mathcal{F} = \{f_1, f_2, \ldots, f_{|\mathcal{F}|}\}$.
- The *background pixels*: the pixels that belong to the background. The values of these pixels are denoted by the set $\mathcal{B} = \{b_1, b_2, \ldots, b_{|\mathcal{B}|}\}$.
- The *don't care pixels*: some pixels are not taken into account in the calculation of the measures. These are for instance the pixels in a close neighborhood to pixels belonging to $\mathcal{F}$, because for these pixels it is unsure whether they are part of the EP catheter or not. Also, pixels belonging to other line structures in the image that are very similar to the EP catheters are not taken into account, because the image processing methods were not designed to distinguish these features (they are distinguished during the last step, see Chapter 5).

In all the tests in this section, the test images are first preprocessed using the background equalization method in Section 3.3. The parameters of this step are determined using visual inspection of the result. The values are fixed to $\sigma_b = 1$ and $d_{\text{be}} = 17$ pixels.

6.1.2 Evaluation measures

The properties in the feature images that we find most important are:

1. Strength of the response of the feature of interest (the EP catheter) compared to the background.

---

[1] Note that images in the two sets with equal values for mAs, can not be considered the same image quality, because the settings during both acquisitions were not the same (i.e. other angles, other distances, and for the acquisition of EP2 additional perspex plates were used).

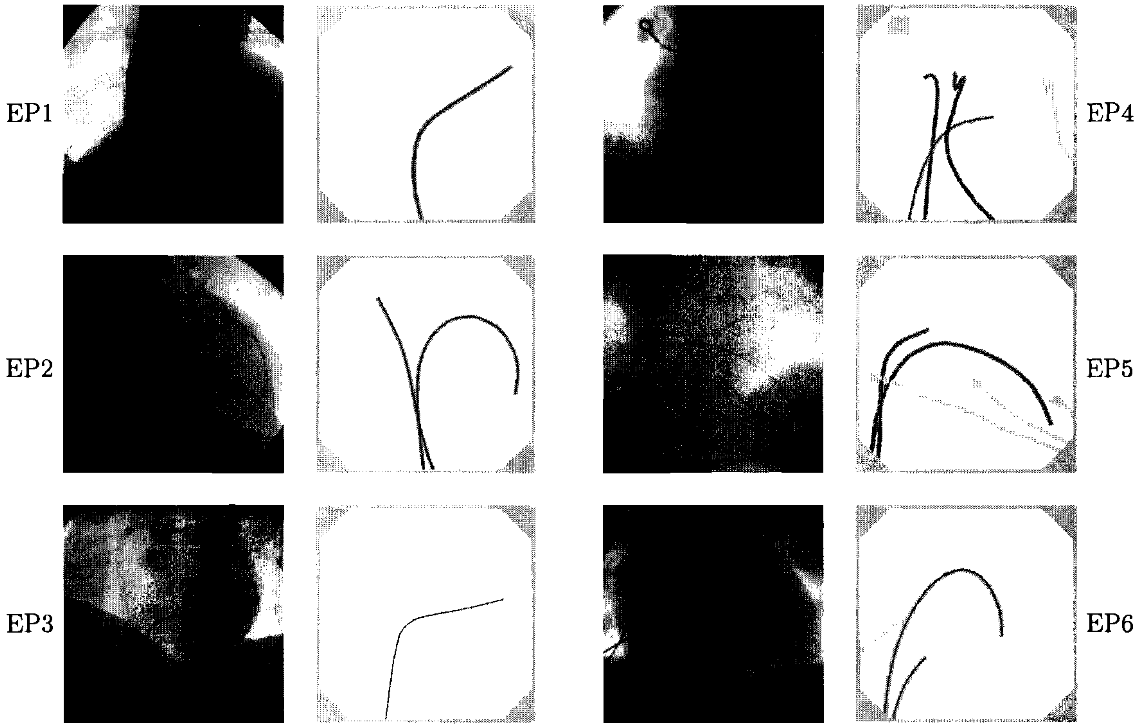


Figure 6.1: *Images used for evaluation of feature image quality, with ground truth. In the ground truth images, white means background, gray means "don't care" and black means EP catheter.*

2. Consistency of the response. For instance, if there are gaps in the lines representing the EP catheters, or if there are a lot of strong responses in the background, the response is less consistent.

Because it is hard to find a measure that combines these two effects, we introduce two separate evaluation measures.

The signal-to-background ratio (SBR)

The *signal-to-background ratio* (further on abbreviated as "SBR"), which is also used in [28], is defined as

$$\mathrm{SBR} = \frac{\overline{\mathcal{F}}}{\overline{\mathcal{B}}}, \tag{6.1}$$

where $\overline{\mathcal{F}}$ and $\overline{\mathcal{B}}$ are the averages of all pixel values in the sets of foreground resp background pixels (i.e. $\overline{\mathcal{F}} = \frac{1}{|\mathcal{F}|}\sum_{i=1}^{n} f_i$).

The histogram-consistency (HC)

Suppose we would apply a threshold of value t on the feature image, and consider every pixel with value higher than t as part of the catheter and pixels with lower value than t as part of the background. There will be false positives (i.e., pixels with value higher than t, but not part of the EP catheter), and false negatives (i.e., pixels with lower value than t, but part of the catheter). Commonly used measures to express this are the *sensitivity*

$$\text{sensitivity} = \left(\frac{1}{|\mathcal{F}|}\sum_{i} \varepsilon(f_i - t)\right), \tag{6.2}$$

and the *specificity*

$$\text{specificity} = \left(\frac{1}{|\mathcal{B}|} \sum_j \varepsilon(t - b_j) \right), \tag{6.3}$$

where $\varepsilon(x)$ is the unit step function. In other words, the sensitivity is the amount of false negative pixels relative to the number of pixels that belong to the catheter according to the ground truth. The specificity is the amount of false positive pixels relative to the number of pixels that belong to the background. The *histogram-consistency* (further on abbreviated as "HC") is defined as the product of sensitivity and specificity at the value of t where this product is maximal

$$\text{HC} = \max_t \left(\frac{1}{|\mathcal{F}|} \sum_i \varepsilon(f_i - t) \right) \left(\frac{1}{|\mathcal{B}|} \sum_j \varepsilon(t - b_j) \right). \tag{6.4}$$

This measure gives an indication how well the pixels belonging to the catheter can be separated from the background, if an optimal threshold t would be used. A value of 1 indicates that this can be done perfectly.

6.1.3 Comparison of different local feature detectors

In Chapter 3, several different local ridge detector were introduced. In this evaluation, we will take three different filters, introduced in that chapter, into account:

- The largest eigenvalue of the Hessian matrix. This is the most common ridge detection filter. The filter is further on referred to as *Hessian*. The coefficients of the m-components of this filter are: $\alpha_0 = 1$, $\alpha_{|2|} = 1$ (see Equation 3.44, page 33).

- The filter proposed by Meijering [33], in which a larger weight is assigned to the $|m| = 2$ component, to make the filter more elongated. The filter is further on referred to as *Meijering*. The coefficients of the m-components of this filter are: $\alpha_0 = 1$, $\alpha_{|2|} = 2$.

- The filter proposed in Chapter 3, where an $|m| = 4$ component is added, to make the filter even more elongated. The filter is further on referred to as *Added m4*. The coefficients of the m-components of this filter are: $\alpha_0 = 1$, $\alpha_{|2|} = 2$, $\alpha_{|4|} = -0.72$.

SBR and HC versus local scale

The most important free parameter for the local filters is the scale σ_{local}. Therefore, to get an indication of the performance of the different filter with respect to the SBR and HC, Figure 6.2 shows the resulting measurements for 2 different images as function of σ_{local}. The scale was measured over the range of $\sigma_{\text{local}} = 2$ to $\sigma_{\text{local}} = 8$ with steps of 0.25. All other images in the test set show similar curves.

The observations are:

- In all cases, the SBR for the Hessian filter is best. This can be explained by the observation that for absolute values of higher m-components of a filter, the SBR decreases, because higher m-components pick up higher frequencies and are thus more sensitive to noise. This is clearly visible in Figure 3.11 (page 37): for higher values of m, the catheter is less visible. All filters consist of a linear combination of m-components. The Meijering and Added m4 filter assign heavier weight to higher m-components than the Hessian, so the resulting SBR will be smaller.

- The HC for the Added m4 filter is better. Apparently the Added m4 filter is better capable of consistently separating foreground from background (HC), but with a lower amplification factor (SBR).

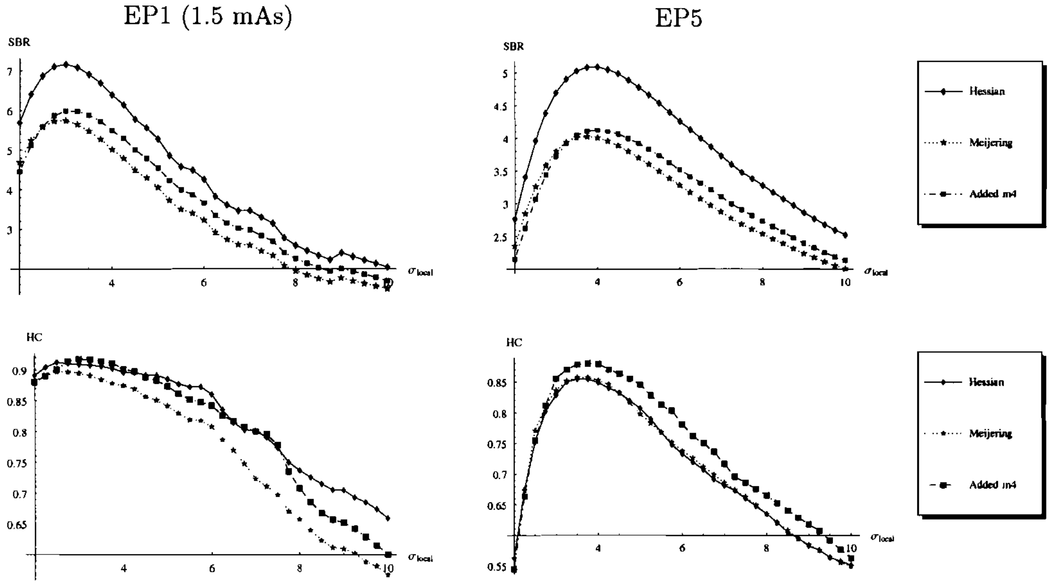


Figure 6.2: *Measured SBR and HC as function of local scale (expressed in pixels) for three different local detectors, on the images EP1 and EP5 (Figure 6.1). See text for details.*

- Both measures show a clear peak for a certain scale. The scale at which both SBR and HC have the peak do not differ very much. In Table A.1 (Appendix A) the measured optimal values for σ_{local} are indicated for the six different images. Taking the average of these values, results in a scale of $\sigma_{\mathrm{local}} = 3.4$ that will be used in further experiments. The common method to determine the optimal derivative scale of a feature as proposed by Lindenberg [29] leads to similar values for the optimal scale.

SBR and HC versus noise

To compare the behavior of the different local filters with respect to X-ray dose level, we measure the SBR and HC on an image that is acquired with different dose levels, and for each image we take the peak response of both measures with respect to scale. For this purpose, we do measurements on images EP1 and EP2, see Figure A.1 and Figure A.2 (Appendix A).

In Figure 6.3, the measured SBR and HC are plotted as function of the logarithm of the beam current I_{beam} (This is proportional to the logarithm of the SNR, i.e. $\log I_{\mathrm{beam}} \sim \log \mathrm{SNR}$, see Subsection 6.1.1). The resulting graphs are shown for the two different images.

- As expected, the values of the measures decrease for lower dose levels.
- The SBR keeps increasing for higher dose levels. The HC curves rapidly increase between 0.2 mAs and 1.0 mAs and are almost constant for higher dose levels.
- The curves for the different types of filters have the same trend, which implies that there is no major difference in noise robustness between the 3 different local filters involved in the test. An exception is the HC of EP1, where the Added m4 filter shows a better performance according to the HC measure.

6.1.4 Added value of tensor voting

To get an indication of the added value of TV on EP catheter images, we also measure the SBR and HC on the output of TV (i.e. on the context-enhanced ridgeness image R). As input, we use the result of the Hessian and the Added m4 filter with fixed optimal parameters, i.e. the values

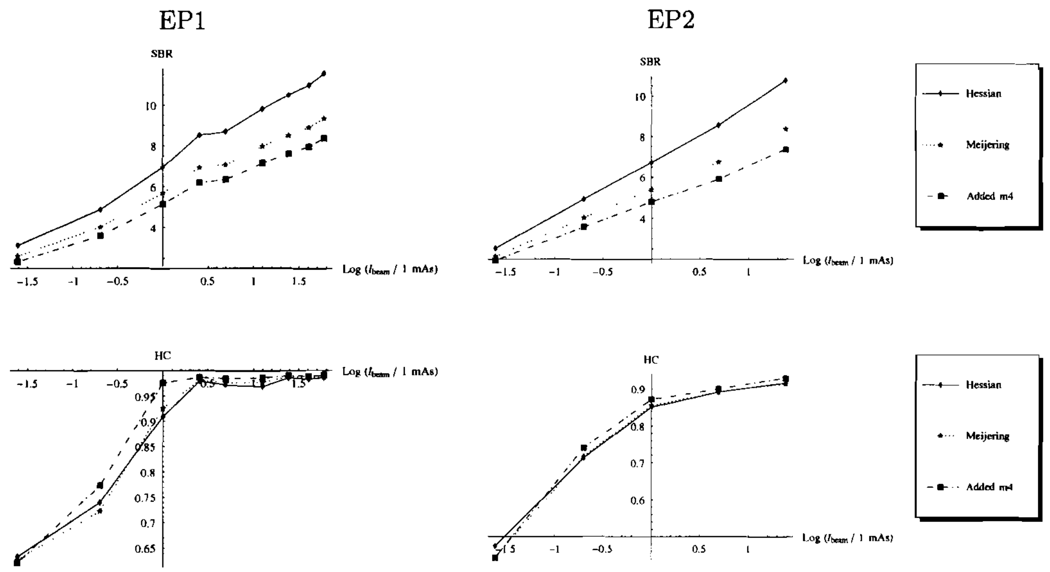


Figure 6.3: *Measurement of the SBR and HC with respect to the noise level of the image. See text for details.*

that were found to be optimal in the previous subsection. We use the steerable tensor voting scheme that was derived in Section 4.3.

SBR and HC versus context scale and number of steerable context field components

In Figure 6.4 the SBR and HC are plotted as function of the context scale σ_{ctx}.

- In most images a gain in SBR and HC is achieved, compared to the SBR and HC for the local ridgeness images.
- Generally a higher optimal σ_{ctx} is found when using the Added m4 filter as input. A possible reason is that the Added m4 filter gives better directional information and therefore we can "safely" broadcast over a larger context.
- The scale at which the SBR or HC is maximum differs quite a lot from image to image (Table A.2, Appendix A). Apparently the characteristics differ too much from image to image. The measures do not give a clear indication for optimal context scale. Nevertheless, by looking at the SBR and HC curves of all images, and at the resulting feature images, we intuitively concluded that $\sigma_{\text{ctx}} = 15$ is a reasonable value in most cases.

During the measurements above, the other free parameter of the voting field, namely n (see Equation 4.25 on page 56), is fixed to $n = 4$. We tried n from 1 up to 4, because larger values result in a number of steerable voting field components that is considered too large. In general, both the SBR and HC measure are best for the highest values of n. This is most likely caused by the fact that a voting field with low n is wider and thus broadcasts more "mess" around, and also because our line structures generally exhibit low curvature.

SBR and HC versus noise after tensor voting

In Figure 6.5 the SBR and HC are plotted as function of the different noise levels of EP1 and EP2 (Figure A.1 and Figure A.2, Appendix A). For comparison, the SBR and HC prior to TV are also shown.

- The SBR and HC are generally larger after TV than before TV.

- An interesting observation in the case of EP1 is that for lower doses, the HC curves after TV are clearly higher than the curves before TV. In this case it seems that TV is capable of maintaining the feature image quality at lower doses, implying that with TV we should be able to handle lower doses.

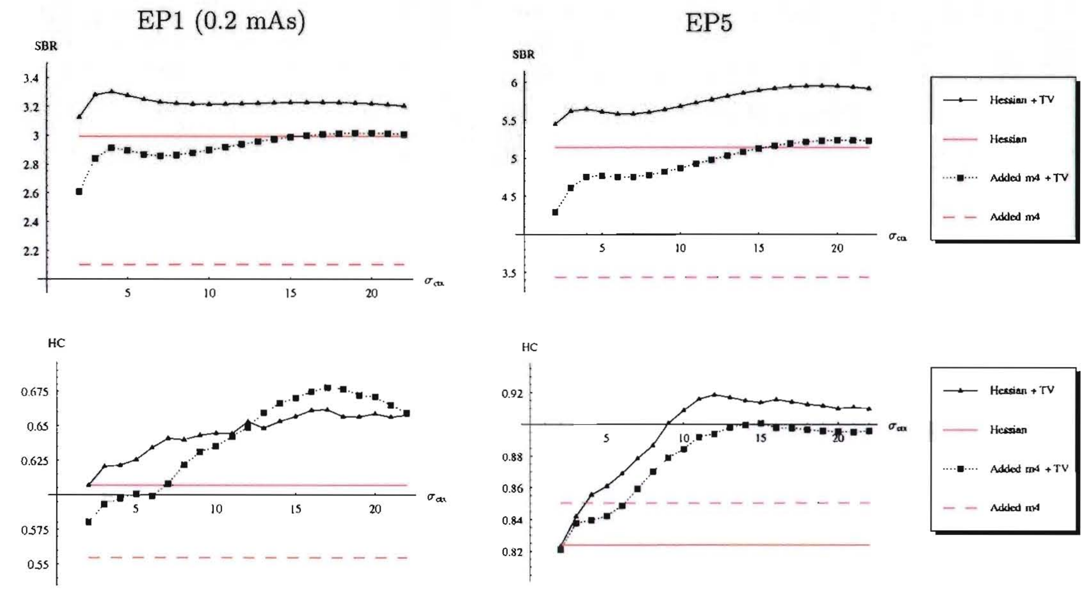


Figure 6.4: *Measured SBR and HC as function of context scale (expressed in pixels) for Hessian filter and Added m4 filter. See text for details.*

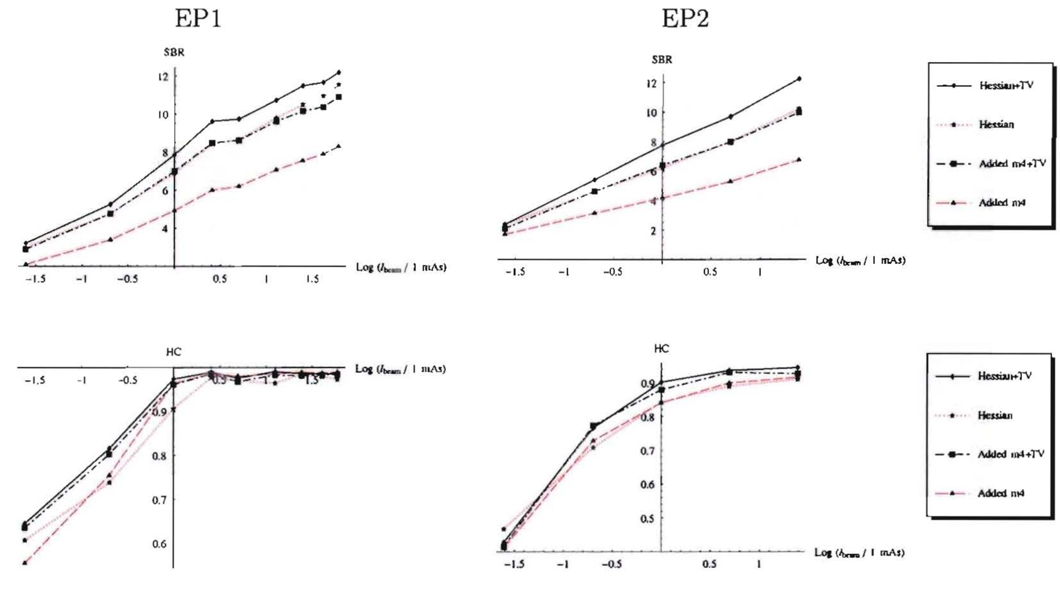


Figure 6.5: *Measured SBR and HC as function of dose levels for Hessian filter and Added m4 filter, with and without the TV step. The settings that were used:* $\sigma_{local} = 3.4$, $\sigma_{ctx} = 15$.

The second tensor voting step

in Subsection 4.4.2 we showed the added value of applying TV a second time, after thinning of the resulting ridgeness image after the first TV step (Figure 4.21, page 63). After thinning of the first TV result, we get a sparse image that only contains nonzero values at local ridge maxima. These local ridge maxima often exhibit a lot of small gaps. The second TV step closes these gaps to make the path extraction easier.

Intuitively, the scale to be used for the second TV step should be smaller than the scale for the first step, because the most important context enhancement is done in the first step and the second step only has to smooth the thinning step. Therefore, we assume for the second context scale $\sigma_{\text{ctx2}} = \sigma_{\text{ctx}}/2 = 7.5$ to be a reasonable value.

6.1.5 Conclusion

In this section, it becomes clear that the Hessian filter performs better than the Meijering filter. It does not become clear whether the Added m4 filter or the Hessian filter performs better. Often the Hessian is better according to the SBR measure, sometimes the Added m4 filter is better according to the HC measure. We did not explicitly evaluate the quality of the estimated orientations. So given the measurement results, we are not able to choose one of these filters. Therefore, in the following we will take both filters into account, to see which one serves best as input for the next step.

For the parameters we found indications for optimal values. The resulting parameters values are indicated in Table 6.1.

Parameter	*value*	Short description	*Determined using*
σ_b	1	Background equalization blur	Visual inspection
d_{disc}	17	Diameter of background equalization structure element	Visual inspection
σ_{local}	3.4	Scale of local filter	SBR and HC
n	4	Voting field width	SBR and HC
σ_{ctx}	15	Scale of voting field for first TV step	SBR
σ_{ctx2}	7.5	Scale of voting field for second TV step	Visual inspection

Table 6.1: *Found optimal parameter values for local feature detection and context enhancement.*

TV generally enhances the SBR and HC for appropriate chosen context scale σ_{ctx}. It seems that using TV leads to better feature image quality, also at lower dose levels.

Of course the evaluation using SBR and HC does not ensure that the found parameter values are optimal in combination with the high-level EP catheter extraction, neither does it ensure that TV indeed leads to a considerable gain in the final EP catheter extraction result. The evaluation of feature images has proven to be difficult. There are a lot of possible things that can go wrong:

- The ground truth might be wrong (the catheter centerlines were indicated manually).
- The evaluation measures do not take all aspects of feature image quality into account. It is not sure if the properties that are important for further use of the feature images, are really contained in the measures.
- The small number of test images might not be representative.
- The approach of step-by-step fine-tuning of parameters could lead to suboptimal parameter values.

Although it has been difficult to draw firm conclusions from the feature image evaluation, we still believe it is good to evaluate the feature images first, instead of only evaluating the end result. At least we now have an indication of parameter values and an indication of the effect of different filters, which reduces the complexity of the optimization of the entire algorithm.

6.2 Evaluation of extraction results

In this section, we will evaluate the final extraction results. The main questions we hope to answer during this evaluation are the following:

- How well are we capable of extracting the tip of the catheter? How about the tip and an additional part? And how about the entire catheter? What are the most problematic situations?
- Is there a clear added value of TV, i.e. is the context enhanced ridgeness image R considerably better than the local ridgeness r, in combination with the EP catheter extraction algorithms? Also, which local feature detector serves best as input?
- What is the robustness to noise, i.e., which dose level gives acceptable results?

To answer these questions, we will perform the following tests:

1. We will qualitatively evaluate the results, by inspecting the results and pointing out the most common problems.
2. We will quantitatively evaluate the extraction results on a set of 50 frames that are randomly picked from the available clinical EP image sequences. These images all have relative good image quality, so measurements will also be performed on the same set of images with additional noise added.
3. To investigate noise robustness in more detail, we evaluate the extraction quality for 5 different noise levels on a set of 12 different images with EP catheters, which are acquired using a thorax phantom.

6.2.1 Evaluation strategy

Segmenting a ground truth for all images, to automatically identify correct extraction results, is time consuming. Alternatively, we visually inspect the extraction results and count the success/failure rates. For the application in mind, i.e. for CNR based dose control, we expect that

- The tip of the catheter is most important for the navigation. Therefore the correct detection of the tip is most relevant.
- The correct identification of the line describing the catheter (tip) is more important than the correct identification of all electrodes. Therefore, if a few electrodes are missed, this is not considered as a significant error.

Instead of only classifying the results in categories "correct" and "failed", we decided to use five different categories (scores) to classify an extraction result. This gives us more information on feasibility of extracting the tip, the whole catheter, etcetera. The categories are indicated by the numbers 0, 1, 2, 3, and 4. In Table 6.2 the categories are defined. Note that there are no explicit requirements on the number of electrodes detected correctly, but only on the part of the catheter containing a certain amount of electrodes. The correctness of extraction is optically checked by looking at the images with and without superimposed extraction result.

If a test image contains more than one catheter, each catheter is categorized separately. The number of catheters in the image is supplied as prior knowledge to the algorithm.

For the quantitative evaluation, a score as defined in Table 6.2 is assigned to all catheters in a set of images. Then, statistics are gathered on the percentage of success and failures, in the following classes:

- %fail: the percentage of catheters with score 0 or 1;
- %frag: the percentage of catheters with score 1;
- %tip: the percentage of catheters with score 2, 3, or 4;

Score	*Description of extraction result*
0	The catheter is not detected at all, the detection result is completely off.
1	The catheter is not detected correctly, but at least 75% of the detected path does coincide with the EP catheter.
2	The tip extraction of the catheter is correct. An extracted tip is considered correct if it covers the part of the catheter containing the real end-point of the catheter, and at least the 4 electrodes closest to the end-point for catheters with 8 or less electrodes, or at least 50% of the electrodes for catheters with more than 8 electrodes. A large ablation tip is counted as two electrodes.
3	The tip part is correct, together with an additional catheter segment with minimal the same length as the minimal required tip length (according to score 2).
4	The entire catheter is detected correctly, that means: one end-point of the curve describing the catheter must coincide with the real end-point of the catheter, the other end-point should have a distance to the closest image border of at most 10% of the total image size, e.g. for an image with size 512 × 512 the maximum distance is 51 pixels.

Table 6.2: *Scores for quantifying EP catheter extraction results.*

- %tip+ext: the percentage of catheters with score 3 or 4;
- %entire: the percentage of catheters with score 4.

The catheters are extracted by our program written in Mathematica and C++. To easily view the extraction results, a Java plugin for ImageJ[2] is written. See Appendix B for more information on the software.

6.2.2 Setting parameters for EP catheter extraction

The EP catheter extraction method of Chapter 5 has a lot of parameters. For practical reasons, therefore, the only workable method was to tune these parameters intuitively and by "trial and error". For that purpose, we used a set of 10 images that are not part of the test sets used further on. The parameters are set such that the extraction works as well as possible on these images. The resulting parameter values are shown in Table 5.2 (page 90). The values were fixed for all further measurements.

6.2.3 Qualitative evaluation of extraction results

In this section we will qualitatively evaluate the extraction results. We will point out common errors, and indicate possible causes of this problems.

The most common mistakes are:

- Especially for catheters with a large number of electrodes, it happens that a number of electrodes on the EP catheter are detected, but the real end-point of the catheter is missed (Figure 6.6a-b). This can have several causes: a path containing electrodes might be missed, the extraction of most salient electrode candidates might miss some electrodes, a connection between neighboring electrodes in the electrode graph is not made (due to, for instance, wrong orientation information), or the correct electrode group is not made or not selected.
- If the electrodes are not visible at all in the image, the EP catheter extraction fails (Figure 6.6c-d). This problem can only be solved using temporal information.

---

[2] ImageJ is a free image processing and analysis tool written in Java, see http://rsb.info.nih.gov/ij/.

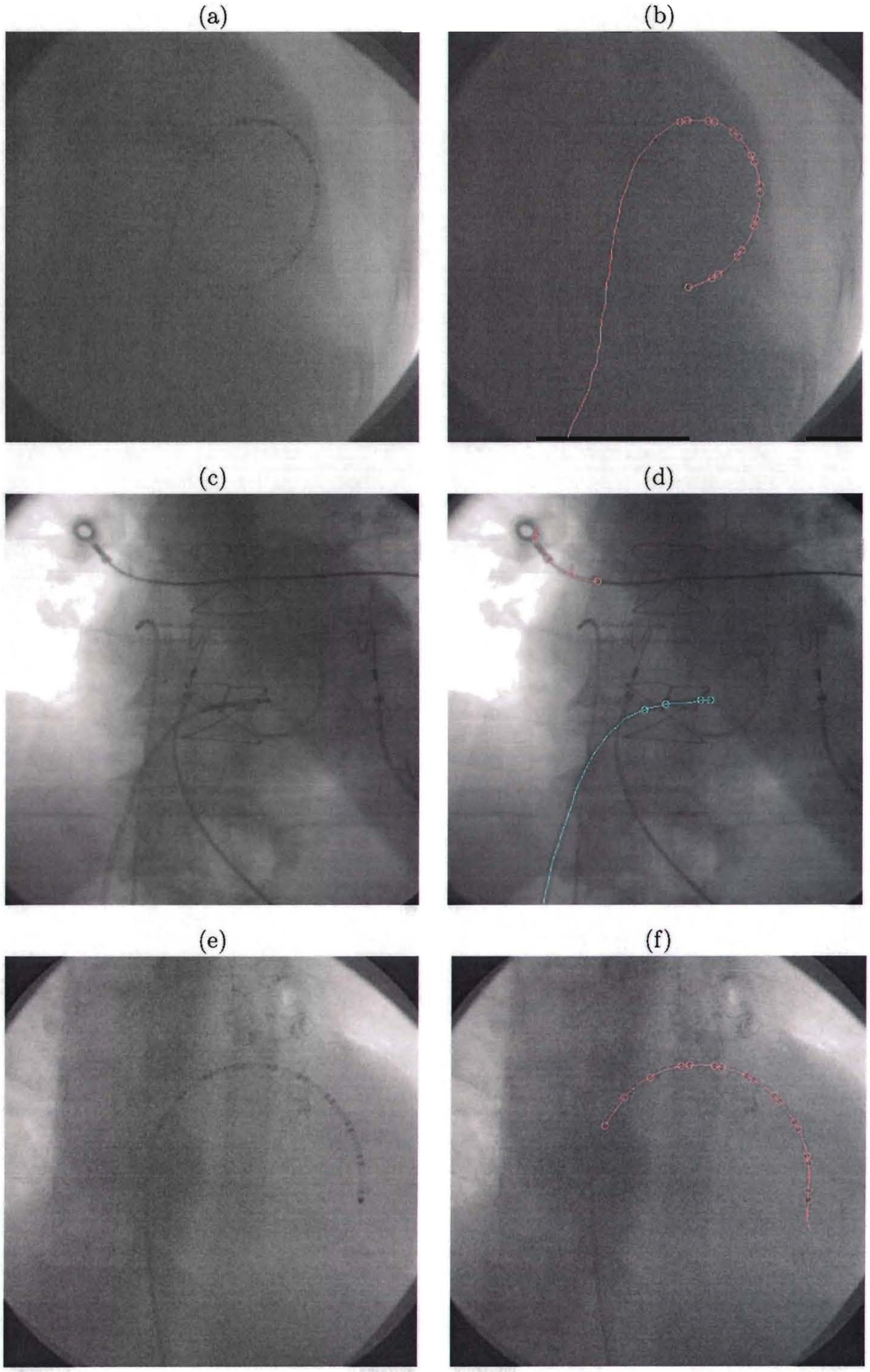


Figure 6.6: *Example extraction results, demonstrating problematic catheter tip detection. (a)-(b) The real end-point of the catheter tip is missed, due to bad visibility (score: 1). (c)-(d) One catheter is detected correctly (except that a lot of electrodes are missed), the other catheter is missed due to bad electrode visibility. Instead, a ECG sticker is detected as catheter (score: 4, 0). (e)-(f) The tip of the catheter is correct, but the wrong extension direction is chosen (score: 2).*

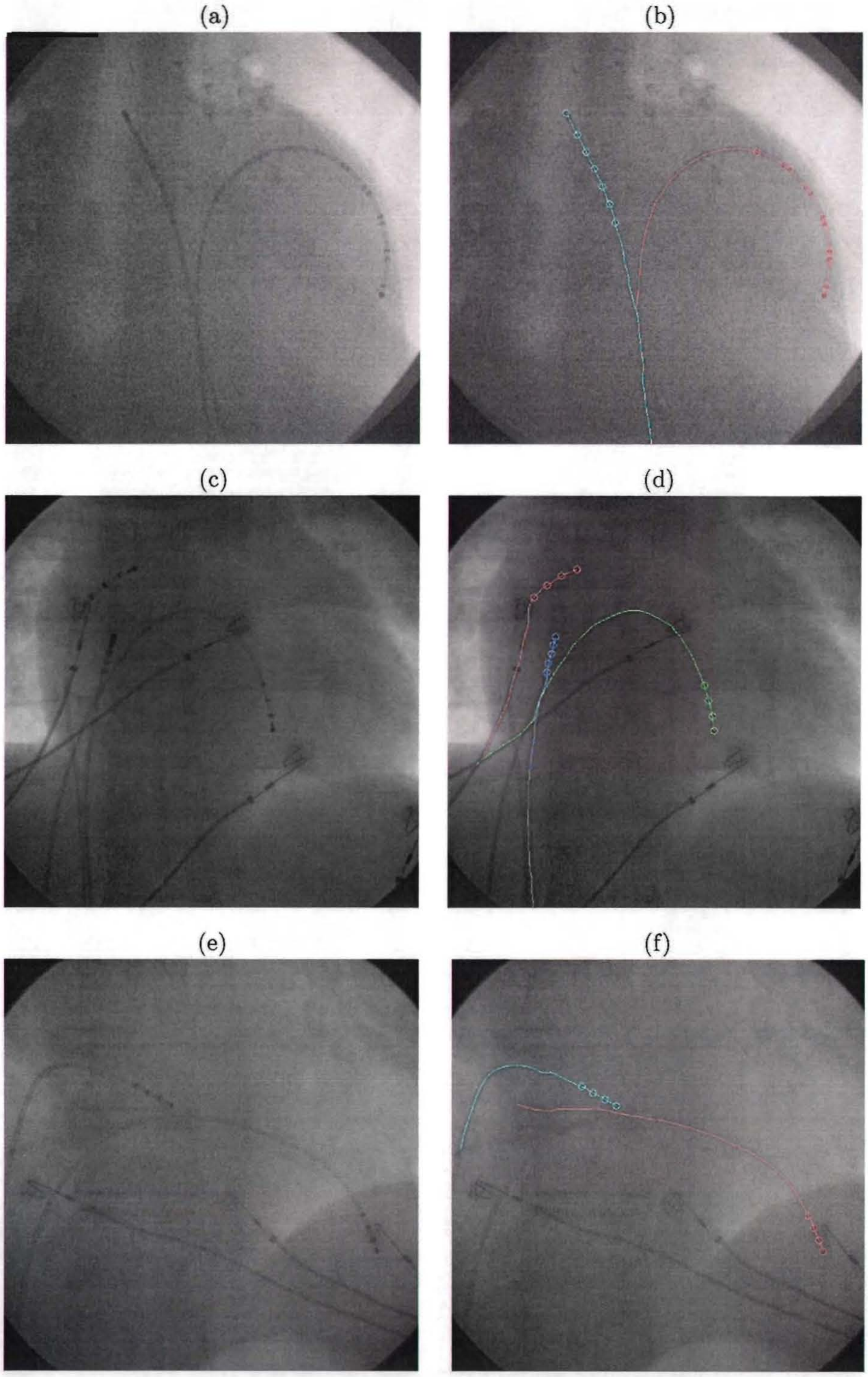


Figure 6.7: *Example extraction results, demonstrating problematic tip extension. (a)-(b) The tips and additional extension of the catheters are correct, but not the entire catheters (score: 3, 3). (c)-(d) One catheter is detected correctly, but two catheters get an incorrect extension due to ECG stickers (score: 4, 3, 2). (e)-(f) One catheter is detected correctly, the other one gets a wrong extension due to a rib (score: 4,3).*

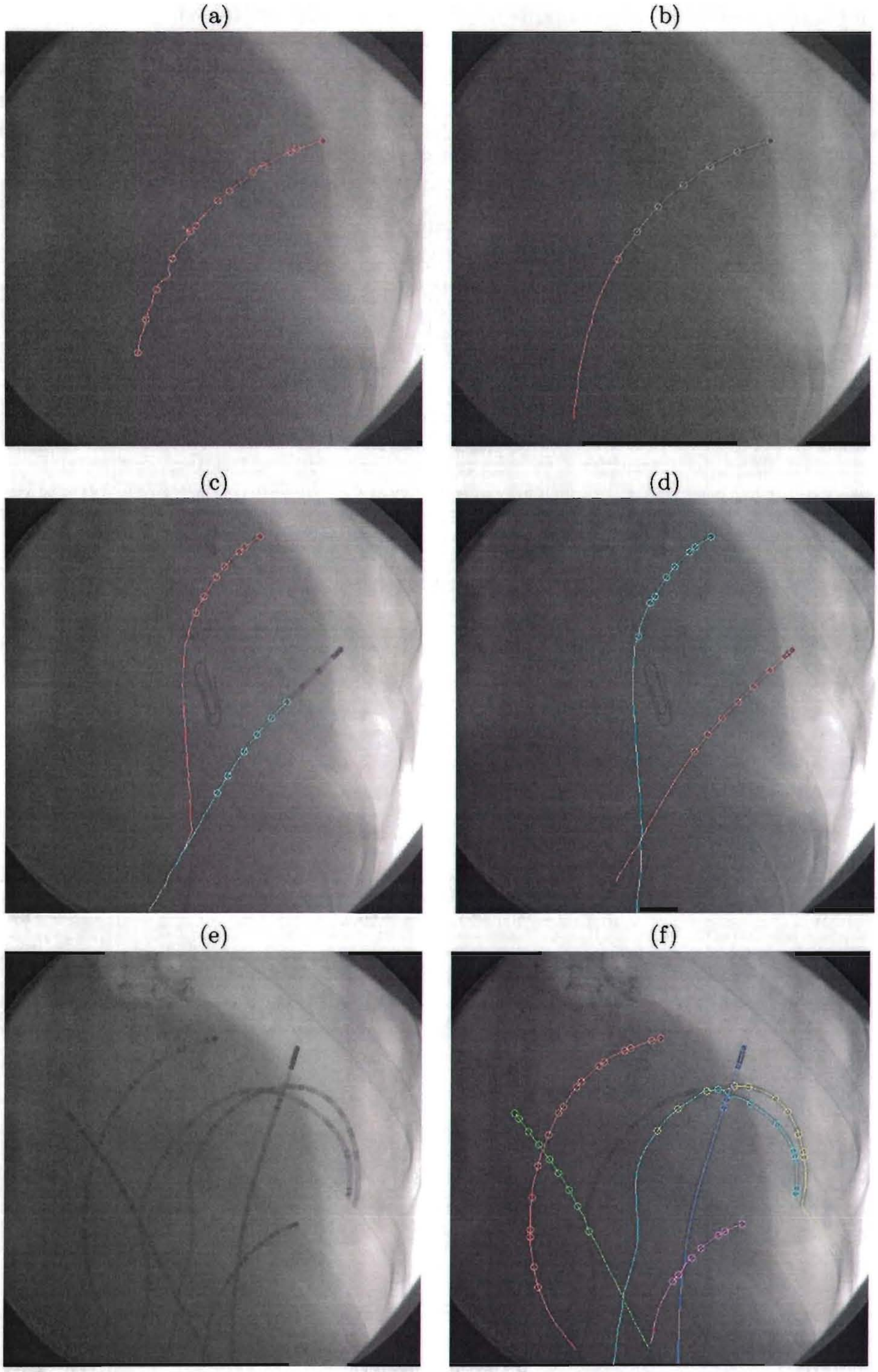


Figure 6.8: *Example extraction results, (a)-(d) demonstrating differences of using catheter extraction with and without TV. (a)-(b) Difference in quality between extraction without (a) and with (b) TV. The result without TV is much less smooth (score without TV: 1 (too messy), score with TV: 4). (c)-(d) Another example in which TV leads to a better result. One catheter is correct in both cases. Without TV (c) a large part of the tip of the other catheter is missed, because part of the catheter tip is not extracted during the path extraction step (score: 4, 1). With TV (d) the ridgeness feature image is enhanced, and therefore the entire catheter is extracted (score: 4, 4). (e)-(f) Example image with 6 catheters. 4 catheters are extracted correctly, and for 2 catheters only the correct tip is extracted (score: 4, 4, 4, 4, 2, 2).*

- Quite often, the decision on the side to extend, starting from the extracted electrode group, is wrong (Figure 6.6e-f). This decision is made by looking which side of the electrode group has the most certain continuation (Subsection 5.3.4). These errors can be caused by other line structures in the neighborhood of the end-point, or because of bad preservation of end-points in case of TV. The value of parameter **`n_PixelsDirectionDecision`** is also critical.

- At crossings between catheters (or other line structures), sometimes the wrong decision is made (Figure 6.7a-b). This is most likely caused by the last step of the extraction process, i.e. the selection of the best super path candidate (Subsection 5.4.2). If we look at Figure 6.7a, it is not surprising that the wrong decision is made. Other line structures that cause problems are ECG stickers (Figure 6.7c-d) and ribs (Figure 6.7e-f). Including temporal information would help here.

- If we compare results with and without the use of TV, we observe that extracted catheters without TV are less smooth than the extracted catheters with TV, because TV smooths and enhances elongated structures (see Figure 6.8a-b). Also, extraction of the tip is sometimes more reliable, because parts of the catheter can be missed during path extraction if TV is not used (see Figure 6.8c-d).

Figure 6.8e-f shows an additional example of EP catheter extraction.

In conclusion, we observe that as long as the electrodes are visible, the tip extraction works quite well. Extraction of an additional catheter segment (score 3 in Table 6.2) works reasonably well. The detection of the entire catheter often goes wrong. In the next two subsections, we will perform a small quantitative analysis on the detection results.

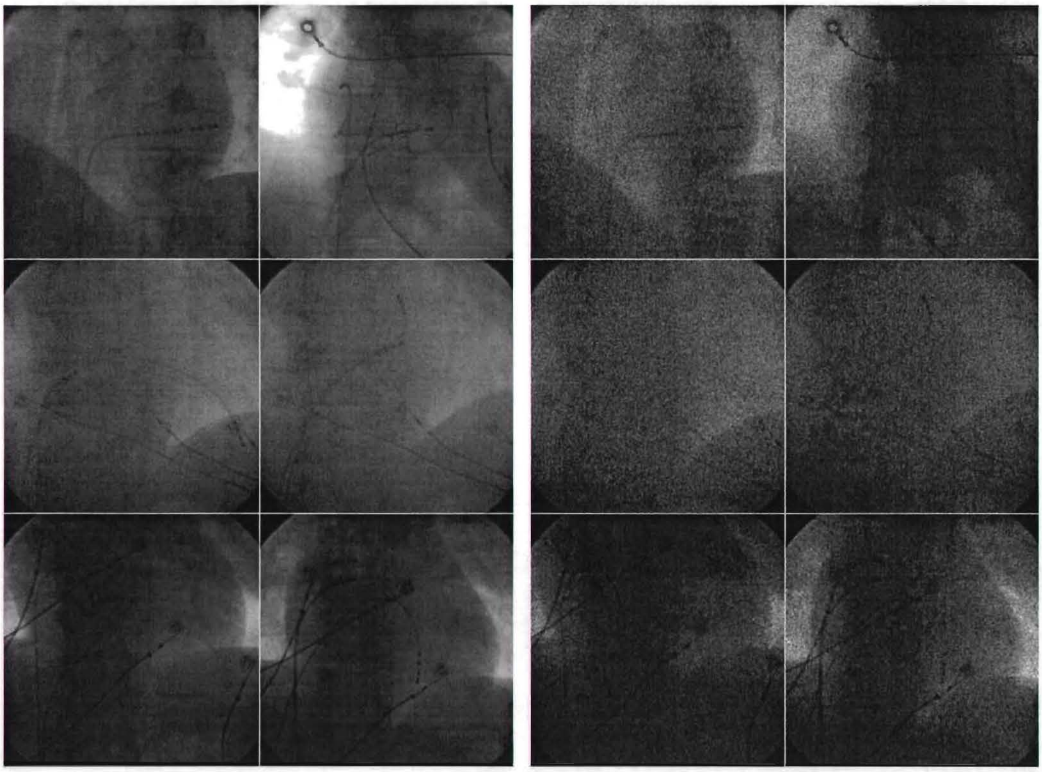


Figure 6.9: *6 examples of clinical test images (out of 50 in the test set), with no additional noise added (left) and additional noise added (right).*

6.2.4 Quantitative extraction results on clinical images

To show the clinical relevance, it is important to see how the method performs on real clinical images. For this purpose, we have image sequences at our disposal from four clinical interventions, all with images showing quite different characteristics (Figure 6.9). In total these sequences consist of hundreds of frames. For this evaluation, we randomly selected 50 different frames from these sequences. Two sequences showed very limited motion, while the two others showed much more motion and therefore more different spatial configurations of the EP catheters. We therefore included 5 frames of each sequence with low motion and 20 frames of each sequence with a lot of motion.

The EP catheter extraction was tested using the following variations for generating feature images:

- The Hessian filter and TV.
- The Hessian filter without TV.
- The Added m4 filter and TV.
- The Added m4 filter without TV.

As prior knowledge, the number of catheters to be found in each image is supplied to the algorithm. In total the 50 frames contain 103 EP catheters.

To check noise robustness on clinical images, we added artificial multiplicative Poisson noise to the images with an average of 40, using RandomJ[3], see Figure 6.9. Although multiplicative Poisson noise does not exactly simulate the nature of X-ray noise, it is a better approximation than Gaussian noise.

For each measurement we calculate the confidence interval that indicates the boundaries, within which a measurement is expected to lie with a probability of 95%. The ensembles of successes and failures exhibit a binomial distribution with probability of success $p = \frac{\% success}{100}$. The probability distribution that we have k successes in a test set of n catheters then is $P(\# success = k) = \frac{n!}{k!(n-k)!}p^k(1-p)^{n-k}$. The standard deviation of this probability density function expressed in percents is $\sigma_{\%} = \frac{100}{n}\sqrt{k(1-\frac{k}{n})}$. A confidence interval CI of 95% is equivalent to taking two standard deviations to both sides, i.e.

$$\mathrm{CI}(n,k) = [-2\,\sigma_{\%}, +2\,\sigma_{\%}] = \left[-\frac{200}{n}\sqrt{k(1-\frac{k}{n})}, +\frac{200}{n}\sqrt{k(1-\frac{k}{n})}\right], \tag{6.5}$$

where n is the number of catheters in the test set, and k is the number of successes.

Figure 6.10 (and Table A.3, Appendix A) shows the extraction results on the set of images. Observations:

- The results *with* TV are a lot better on low noise images, but especially on high noise images. For instance, detection of the tip in the high noise images using the Hessian without TV succeeds in 43% of the cases, while with TV the success rate increases to 72%.
- The use of the Hessian as input filter leads to better performance than the Added m4 filter. This was also observed in the evaluation of the feature images (Section 6.1).

From this observations it can be concluded that TV clearly leads to better results on this set of clinical images.

6.2.5 Quantitative extraction results for different dose levels

For this test we use a set of test images that are acquired at different dose levels, using a thorax phantom and one or more catheters. This is a set of 12 different EP catheter images, each acquired at 5 different dose levels, see Figure A.3 (Appendix A). All these images were acquired with the

---

[3] RandomJ is a plugin for ImageJ, see http://imagescience.bigr.nl/meijering/software/randomj/

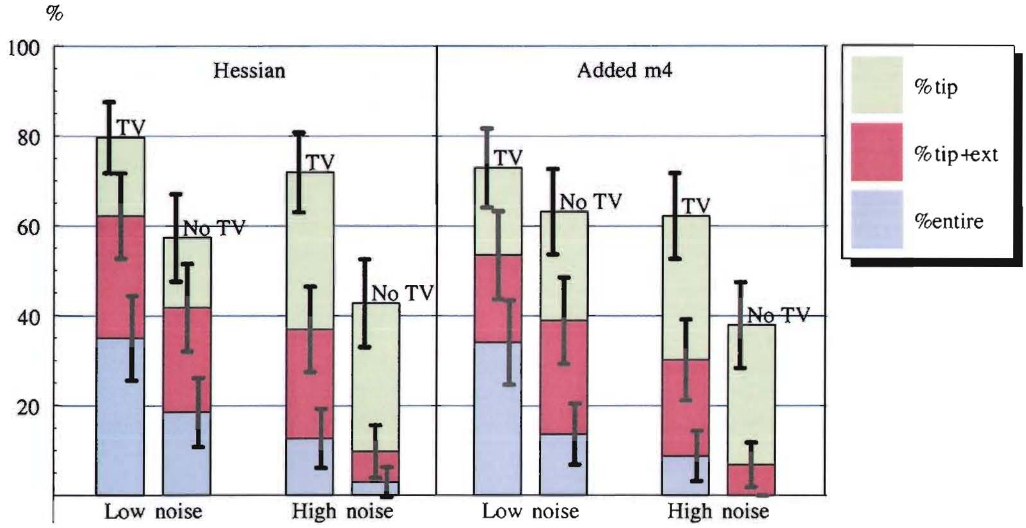


Figure 6.10: *Catheter extraction results for different inputs. The bars with "low noise" are the results of extraction on the original images. The bars with "high noise" are the results of extraction on images with added multiplicative Poisson noise. The bars are subdivided in three parts to indicate successful tip extraction, additional catheter segment extraction, and entire catheter extraction. The gray vertical lines with horizontal serifs indicate the confidence intervals of 95% (Equation 6.5). Note the differences in results with and without TV.*

same X-ray settings, so the image quality should be approximately the same for all 12 images with the same dose level. In total, the 12 images contain 25 EP catheters. Note, that we can not make an absolute comparison of the extraction success rates in this test and in the test on clinical images, because we do not know what dose level in this test is equivalent to the low noise and high noise images in the previous test, since we do not have an absolute quality measure for the input images. We can only compare whether the observations of the two tests are the same.

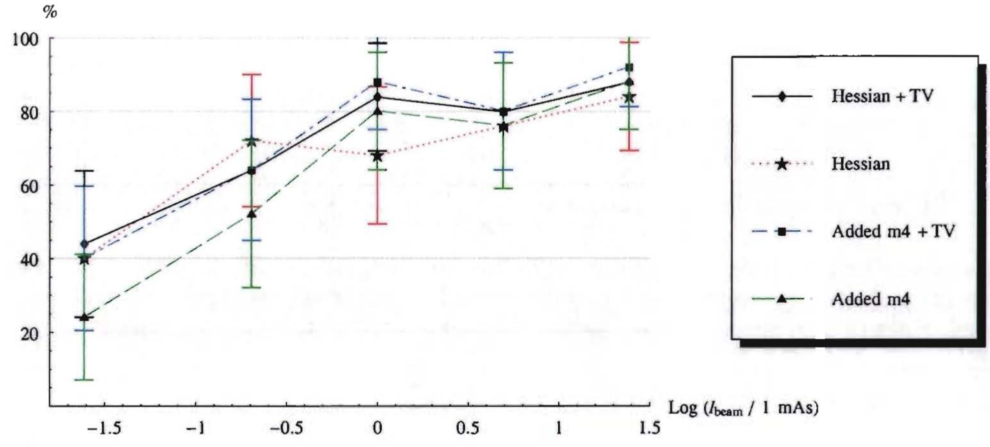
Figure 6.11 (and Table A.4, Appendix A) show the results. Observations:

- As expected the success rate for tip, tip+extension, and entire catheter, increases for higher doses.
- The results after tensor voting seem to be slightly better, especially at 1 mAs. However, taking into account the large confidence intervals, which are caused by the small test set, the differences between the four different input feature images (Hessian+TV, Hessian, Added m4+TV, Added m4) are small.

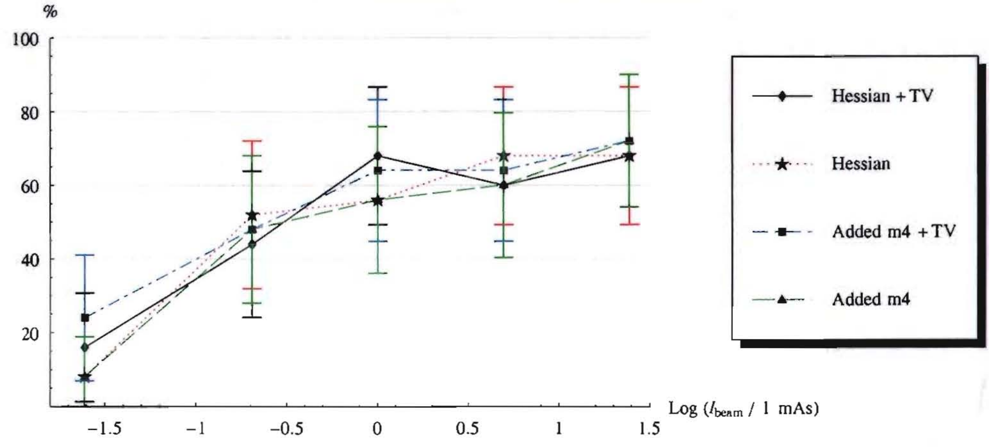
It is not clear why the performance boost with TV on phantom images is so much less than on our clinical images. This is probably caused by a combination of the following:

- Other settings of the X-ray equipment. The phantom images are generally more noisy than the clinical images, which are recorded on a quite high dose. Also, the artificially added noise in the high noise clinical images does not exactly simulate X-ray noise.
- The images in the two test sets have different characteristics, that probably have large influence on the performance of the high-level extraction part of the algorithm (Chapter 5). The most important difference is that the clinical images contain ECG stickers, which are better visible in the X-ray image than the catheters, while the phantom images do not contain other elongated structures similar to EP catheters. The high-level extraction algorithm extracts a *fixed number* (`n_PathExtract`) of paths from the ridgeness feature image. With the use of TV, these paths are typically longer, due to the enhancement of elongated structures

(a) - Extraction of tip



(b) - Extraction of tip and extension



(c) - Extraction of entire catheter

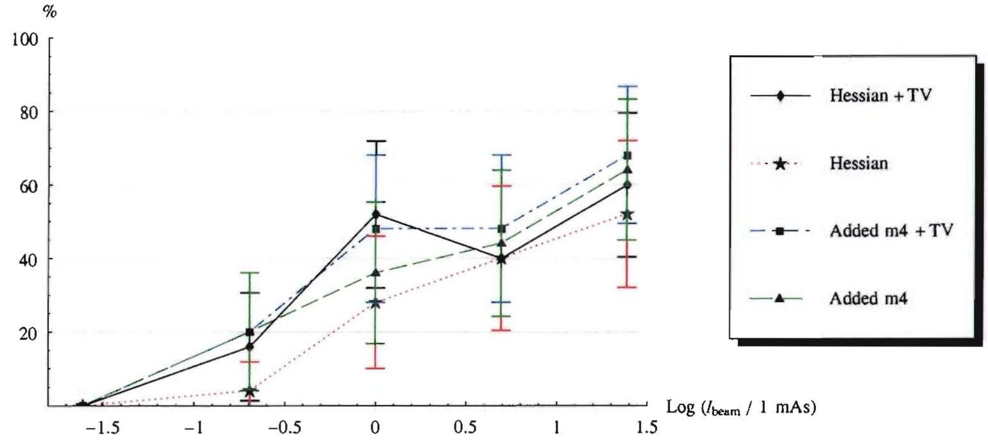


Figure 6.11: *Catheter extraction results as function of dose level (0.2 mAs, 0.5 mAs, 1.0 mAs, 2.0 mAs, 4.0 mAs) on the test set with phantom images, acquired at different dose levels. The vertical lines with horizontal serifs indicate the confidence intervals of 95% (Equation 6.5).*

achieved by TV. So, skipping TV leads to a higher chance of missing parts of EP catheters especially if other line structures are present that are better visible than the catheters.

In conclusion, the test set of phantom images is too small and the images are too different to really draw conclusions from this test. For a more thorough test, it is recommended to use more images, and make the situation more realistic by also using ECG stickers during acquisition.

6.3 Computational issues

This section is a small feasibility study on the computation aspects. First, more detail is supplied about the current implementation, then some speed measurements are performed, and finally some possible ways to gain speed are proposed.

6.3.1 The implementation

The current implementation is written in Mathematica (Mma) and partly in C++. The functions in these two different programming languages interface with each other using MathLink. The reason to write some functions in C++, is that some types of algorithms are very slow in Mma, e.g. functions that perform non-standard pixel-wise operations. Also, operations with graphs and other complicated data structures are cumbersome to program in Mma. The entire high-level extraction was therefore implemented in C++, with extensive use of the C++ STL (Standard Template Library).

The package *MathVisionTools*[4] is used to calculate the Gaussian derivatives. No other non-standard Mathematica packages are used[5]. In C++, no non-standard libraries are used, except for MathLink. Some code was taken from Philips Medical Systems (the Euclidean distance transform). More information on the software can be found in Appendix B. In Table 6.3 the different main steps of the algorithm are summarized.

6.3.2 Speed measurements

To get an indication of the speed of the current implementation, we perform some speed measurements. The speed is measured for the different steps, on 20 random EP images. For this timing measurements the Mma command `AbsoluteTiming` was used, which measures the real elapsed time. The computer is an Intel Pentium 4, 2.26 GHz, with 1 GB of memory, running Windows 2000, without other heavy processes running in the background.

Table 6.4 shows the resulting average time measurements (in the column $q = 1$). Observations:

- As expected the TV step takes most time, i.e. more than 50% of the total, because TV consists of a number of convolutions with a fairly large set of complex-valued two dimensional kernels. However, we already achieved a considerable speed gain with the use of steerable tensor voting method (Section 4.3), compared to an earlier C++ TV implementation that did not exploit steerability.
- The local feature detection step also takes a relative long time, while these operations should not be extremely expensive. For the Hessian case, measurements on the substeps showed that this is mainly caused by the step that calculates the maximum ridgeness and orientation (i.e. a step consisting of only pixel-wise operations). This is probably caused by an inefficient way Mathematica executes this calculations. The difference between Hessian and Added m4 is caused by the fact that for the latter case, more operations are required to steer the filter to the maximum orientation (i.e., a pixel-wise Newton-Rhapson optimization process is used).
- Compared to the other steps, the high-level catheter extraction step is almost negligible.

---

[4] See http://www.bmi2.bmt.tue.nl/image-analysis/Research/Software/Mathematica/AddOns/MathVisionTools/index.html

[5] There exists an image processing package for Mathematica, but this did not contain the desired functionality. For instance, it does contain morphological operations, but these functions are extremely slow.

Step	*Operations*	*Order*	*Implementation*
Background equalization	Closing (dilation+erosion) with structure element of pixel size 17 × 17, pixel-wise division	$\mathcal{O}(d_{\text{be}}^2 s^2)$	Mma, C++
Gaussian derivatives	Convolutions with x,y-separable kernels	$\mathcal{O}(\sigma_{\text{local}} s^2)$	Mma[a]
Calculation of feature images r, ψ, and b	Pixel-wise calculations	$\mathcal{O}(s^2)$	Mma, C++[b]
Steerable tensor voting	Set of convolutions with complex numbers	$\mathcal{O}(\sigma_{\text{ctx}}^2 s^2 c)$	Mma
Thinning of context enhanced ridgeness	Pixel-wise comparison with two pixel neighbors	$\mathcal{O}(s^2)$	C++
Steerable tensor voting	Set of convolutions with complex numbers	$\mathcal{O}(\sigma_{\text{ctx2}}^2 s^2 c)$	Mma
Path extraction	Thinning, extraction of pixel string, path splitting	[c]	C++
Electrode extraction and grouping	Blob extraction, electrode graph, electrode grouping	[c]	C++
Path grouping	Creation of catheter tip paths, searching best extension	[c]	C++

[a] Package `MathVisionTools` is used for this purpose.

[b] Calculation of maximum response for Added m4 filter is implemented in C++, the rest in Mma.

[c] The order of these steps is hard to specify, because the steps consist of a lot of substeps. Also, these algorithmic steps do not run in constant-time, i.e. the time it takes is dependent on parameter values and image content, meaning that the order will only tell something about the worst cases computation times.

Table 6.3: *The steps of the algorithm, their algorithmic order, and the way they are implemented. The parts written in C++ are called from Mma (Mathematica) using MathLink. In the column "order", s is the image size in one dimension, d_{be} is the size of the structure element for background equalization, σ_{local} is the local scale, σ_{ctx} and σ_{ctx2} are the context scales for the first resp second tensor voting step, and c is the number of steerable components of the voting field. The column "Implementation" shows in which language the step is implemented.*

6.3.3 Possibilities for speed improvements

There are a lot of possibilities to improve the speed of the algorithm. First of all, an implementation entirely in C++ would lead to a high gain. C++ is not only faster, but it also saves the overhead caused by the communication between Mathematica and C++. Furthermore, the algorithm could be made faster, in the following ways:

- Using smaller data types to store the images and other data. Currently, a single pixel of an image is stored as a C `double`, which takes 8 bytes, because this is the way Mathematica stores the data internally. Using smaller data types will lead to a considerable gain in speed and a decrease in memory requirements.

- Sub-sampling of the input image. For instance, sub-sampling with a factor 2 will already lead to a time gain of a factor 4 for the most time-consuming steps. However, the extraction quality might decrease because detail is lost. In the next subsection, some experiments will be performed on this idea.

- The large convolutions in the steerable tensor voting step could become faster if implemented via the Fourier domain. These large convolutions could even be calculated using specialized hardware, e.g. a GPU (Graphics Processor Unit) is often faster for convolutions. Convolutions are also very suitable to be implemented in parallel.

- The input of the second TV step is sparse. A TV implementation can exploit this by skipping tensor broadcast at zero locations. Our C++ TV implementation is indeed sometimes a bit faster on very sparse data than the steerable tensor voting implementation.

- The use of temporal information can lead to a gain in speed. For instance, the search area can be restricted.

	Average time		
Step	$q = 1$	$q = 2$	$q = 3$
Background equalization	2.3 s	0.3 s	0.1 s
Local feature detection (Hessian)	7.4 s	1.8 s	0.9 s
Local feature detection (Added m4)	12.0 s	3.0 s	1.4 s
Tensor voting	23.9 s	6.3 s	4.0 s
Catheter extraction	1.5 s	0.3 s	0.1 s
Total time (with Hessian)	35.0 s	8.7 s	5.1 s
Total time (with Added m4)	39.7 s	9.9 s	5.7 s

Table 6.4: *Timing of the different steps of the algorithm, for different sub-sampling factors* q*, where* $q = 1$ *means no sub-sampling. The indicated timings (in seconds) are the averages of measurements on 20 random EP images. The measurements represent the number of real seconds that have elapsed. Measured on an Intel Pentium 4 2.26 GHz with 1 GB of memory, running Windows 2000 (without other heavy processes running in the background).*

Although the high-level extraction part does not seem to be the bottleneck in the algorithm, it should be noted that this step could become the bottleneck. The problem with this step, especially for real time requirements, is that it does not run in constant-time: The speed is content and parameter dependent. For instance, at several steps in the algorithm different candidates are created (e.g. electrode groups, tip extensions). In specific situations, and with specific parameter settings, the number of candidates can explode, resulting in an unexpected long processing time (but not infinitely long, because the algorithms are in principle designed such that they will never be caught in an infinite loop). One could calculate the worst case scenario, but this will not tell us much about the processing time in practice. Instead, one could measure the average and variance of the processing time.

6.3.4 Sub-sampling the image

An interesting way to gain speed is to sub-sample the image, prior to detection of EP catheters. The resulting extraction will be less precise. If necessary it can be refined afterwards using the original image. In [6], this approach is used for guide-wire detection. Breitenstein et al. sub-sample the image in such a way that dark line structures are best preserved. This is achieved by calculation of the morphological erosion of the original image with a square structure element of size $q \times q$, where q is the integer sub-sample factor, prior to sub-sampling. The speed gain using this approach is large. Table 6.4 shows the achieved speed gain with the use of sub-sampling.

We perform a small test on sub-sampling the image, to see the change in extraction success rate. The same algorithm was used, but all parameters related to image dimensions were divided by sub-sampling factor q. Tests showed that the extraction quality was only reasonable for a sub-sampling factor of $q = 2$. Higher values for q led to very bad results, because then important structure is lost, and the filters introduce errors because the scales are too small compared to the pixel grid.

Figure 6.12 shows the results for sub-sampling with $q = 2$ compared to no sub-sampling, on the test set of 50 clinical images. We observe that the decrease in extraction performance is high, especially for the noisy images.

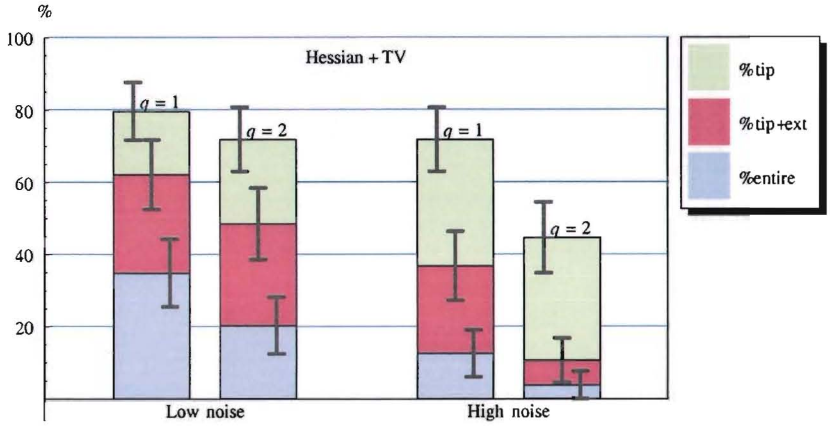


Figure 6.12: *Catheter extraction results for Hessian and TV, with and without sub-sampling with a factor 2. The bars with "low noise" are the results of extraction on the original images. The bars with "high noise" are the results of extraction on images with added multiplicative Poisson noise. The bars are subdivided in three parts to indicate successful tip extraction, additional catheter segment extraction, and entire catheter extraction. The gray vertical lines with horizontal serifs indicate the confidence intervals of 95% (Equation 6.5). Note the differences in performance with and without sub-sampling. Especially for noisy images, the decrease in performance is high.*

Chapter 7

Conclusions and Recommendations

7.1 Conclusions

In this thesis we proposed methods for the extraction of EP catheters using spatial information only and without the need for a manually pointed initial seed. We used a three step approach to accomplish this.

We first detected local image features (ridges and blobs) using the differential structure of the image. Based on a model of a line in an image, we derived steerable filters for detecting ridges that become more elongated when taking a larger number of steerable components into account.

Then, we enhanced the elongated structures in the image with the use of a newly derived spatial context enhancement scheme that we call *steerable tensor voting* (TV), which is based on the tensor voting method described in literature. It performs tensor voting with the use of a number of complex-valued convolutions. On dense tensor fields, this method is more efficient than the standard way tensor voting is implemented. We proposed a mechanism for *repeated* tensor voting. Applying tensor voting twice, with a thinning step in between, proved to be helpful for further enhancing elongated line structures.

Finally, we introduced a new high-level extraction algorithm for the final extraction of EP catheters. The most salient line structures and blobs are extracted out of the image. These so-called *paths* and *electrode candidates* are used for further grouping, to finally obtain the EP catheters.

In the evaluation, we compared the resulting feature images and final extraction results, for different local feature detectors, with and without the use of the steerable tensor voting step. We tried different local ridge detectors as input, since we expected that more elongated filters would lead to more consistent orientation information, which should be helpful in the next processing steps. However, the standard Hessian ridge detector (i.e. the first eigenvalue of the Hessian matrix) generally proved to give the best results. The reason is probably that the more elongated local ridge detectors render more artefacts in the resulting feature images, since the higher m-components pick up more noise. Larger filters also have a higher chance of picking up wrong structures. Therefore, for now we conclude that the use of the Hessian filter is preferable.

According to the evaluation of both the ridgeness feature images and the final extraction results on clinical images, the use of (steerable) tensor voting leads to better results. In most cases, the signal-to-background ratio and histogram-consistency measures both improved after steerable tensor voting. For the final extraction on clinical images (on a set consisting of 50 images), the success rate of tip extraction improved from 57% to 80% for low noise images. For high noise images it improved from 43% to 72%. Similar gains are achieved for the extraction of the tip with additional catheter segment and extraction of the entire catheter. We also observed that the extracted catheters were much smoother with the use of TV. From this we conclude that tensor voting makes the EP catheter final extraction step easier.

The evaluation results on a set of 12 phantom images, each acquired at 5 different dose levels, were less convincing. The use of different local ridge detectors, with and without TV, did lead to much differences. Probably this is caused by other characteristics of these images: different types of catheters, different X-ray imaging system settings, and the lack of other elongated structures in the images. The results on clinical images, however, are considered more important, so we still believe that the (steerable) tensor voting technique is promising.

We evaluated the performance of our algorithm concerning detection of the catheter tip, the tip with additional extension, and the entire catheter. For Hessian and tensor voting, which gave the best results, the tip extraction on clinical images has a success rate of 80%/72% (low noise/high noise), for the tip with extension the success rate is 62%/37%, and for the entire catheter the success rate is 35%/13%. The results are probably not yet sufficient for use in a final product, but we believe the results can be further improved, see the recommendations. Nevertheless, the results on the tip extraction, which is considered most important for the application, are already quite good.

The computational demands of the algorithm are still far too high for real-time use. We suggested ways to speed up the algorithm. For instance, sub-sampling of the image leads to a considerable speed gain, but especially for the high noise images the performance showed to decrease.

In this thesis, we focussed on a single application, i.e. the detection of EP catheters. We believe, however, that a lot of techniques described in the thesis are applicable for other applications. Many medical applications have a demand for the detection of line structures in noisy images. Two examples of other applications with line structures are discussed in Appendix C. Eventually, the methods can be extended to 3D for detection of lines and surfaces in noisy 3D images.

7.2 Recommendations

The recommendations are divided in two categories: recommendations concerning the practical application of detecting EP catheters and recommendations concerning the theoretical, methodological, aspects.

7.2.1 Recommendations on the EP catheter detection problem

- The most important recommendation is *the use of temporal information.* This will solve a lot of problems: ambiguity at crossings of EP catheters and ECG stickers, invisible electrodes, and wrong decisions concerning the direction, in which to extend a catheter starting at an electrode group. The expectance is that a performance gain and a speed gain will be achieved. Some suggestions:
    - The search area can be restricted by only searching in a certain neighborhood around the catheter position of the previous frame.
    - The certainty measure of the features of Chapter 5 can be adapted such that they get penalized for having a large distance to the extracted catheter positions of the previous image. In this way, the algorithm will prefer features that are close to features in the previous frame.
    - The knowledge that EP catheters move while the ECG stickers stay in place can be used by including this in the certainty measure for catheter (tip) candidates. Also, the heart beat can be used to compensate for the heart beat motion.
    - Once catheters are detected with high certainty in an initial frame, a temporal tracking method can be used, which takes the result of the previous frame as an initial guess. For instance, a spline can be fitted on the ridgeness feature image [4].
    - Tensor voting can be performed in 3D, where the time axis is the third dimension. This can lead to even more enhanced line structures in the images, but the computational load will be high.
- *Optimize the parameters* of the algorithm. There are a lot of parameters, which probably can be further optimized by using statistics on a larger set of test images.

- Create a *more efficient implementation*. An implementation entirely in C++ will be a lot faster. Also, it is expected that using specialized hardware will lead to a considerably gain in speed on, for instance, the steerable tensor voting step. See Section 6.3.

- The goal is to embed the EP catheter detection in a Contrast-to-Noise Ratio (CNR) based fluoroscopy imaging system (see Figure 2.6, page 19). To accomplish this goal, the next steps are:
  - Measure the CNR, using the EP catheter extraction result.
  - In our evaluation we investigated the noise robustness of the extraction results for different beam currents. This merely gives an indication on noise robustness. To know how well it will work in the CNR control system, the algorithm performance needs to be evaluated with respect to the CNR. So, it should be investigated for what CNR the EP catheter extraction still works.
  - Investigate the feasibility of the CNR measure if it is calculated using the tip of the catheter, the tip and additional catheter segment, or the entire catheter.

  In [34], methods to measure the CNR and to control the dose are explained in detail.

7.2.2 Recommendations on the theoretical aspects

On local feature detection and context enhancement (see also Sections 3.8 and 4.5):

- The local filters can probably be improved by choosing a better justified set of basis functions, and more advanced criteria [20].

- Include curvature in the line structures, in the local filters and/or the voting fields. Now, we only use the estimated orientation to align the voting field. We expect that if the voting field is adapted to the curvature, we obtain better context enhancement. It would be interesting to investigate the possibilities of making *steerable filters with respect to curvature*. This is a lot more complicated than steerability with respect to orientation. We need a group whose group action has the effect of "bending" the kernel. This will necessarily be a non-linear group. This is still rather speculative, and we do not know if this could lead to a feasible method.

- Investigate the use of different communication mechanisms. As we mentioned in Chapter 4, we use a *broadcasting* mechanism to communicate with the context, where the voting is achieved by tensor addition. More sophisticated communication mechanisms are possible. We believe that a kind of combination of collection and broadcasting could be better.

- Currently, we only vote with $|m| = 2$ components. This results in bad preservation of end-points and crossings of line structures. Including other $|m|$ components will lead to better results. Especially the $|m| = 1$ component is important for preserving end-points [38].

- Design of a voting field using a more sophisticated model or using statistics of lines in images.

- Multi-scale tensor voting. Tensor voting has a context scale parameter. The meaning of this scale parameter is different than the local scale parameter. A larger context scale removes more fine details and is thus more appropriate for smoother line structures. If the characteristics of the line structures in an image are unknown, a multi-scale approach is useful. It is not yet known how to embed tensor voting in a multi-scale framework.

- Link with other techniques. Tensor voting clearly has aspects in common with, for instance, geometry-driven diffusion and diffusion tensor imaging. Comparing methods proposed in these areas can lead to better insights and eventually better methods.

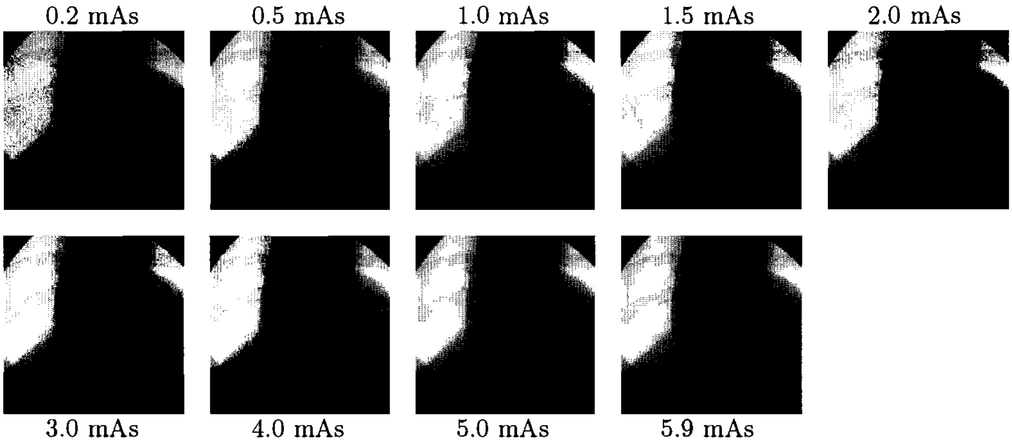
On the high-level extraction (Chapter 5):

- Our approach for high-level extraction is largely application-specific. From the theoretical point of view, general applicable algorithms are more interesting. Interesting publications on generic high-level perceptual grouping are [3, 10].

Appendix A

Measurement Results

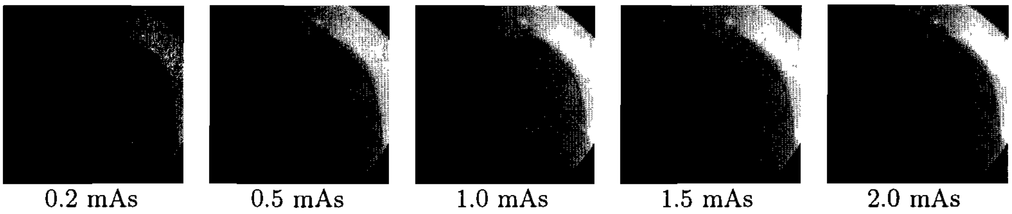
A.1 Evaluation and tuning of feature image quality



0.2 mAs 0.5 mAs 1.0 mAs 1.5 mAs 2.0 mAs

3.0 mAs 4.0 mAs 5.0 mAs 5.9 mAs

Figure A.1: *Different recorded dose levels of EP1. To get realistic noise, the same image was recorded with different X-ray settings. A thorax phantom was used to make the situation as realistic as possible. In all images, the X-ray beam was tuned to 72 kV, continuous fluoroscopy, and a 0.5 mm thick copperplate was attached to the X-ray detector. The beam current was adjusted to vary the image quality, as indicated for each picture. The images were recorded on 28 April 2004 at Philips Medical Systems, Best.*



0.2 mAs 0.5 mAs 1.0 mAs 1.5 mAs 2.0 mAs

Figure A.2: *Different recorded dose levels of EP2. The same procedure was used as for EP1 (Figure A.1). In all images, the X-ray beam was tuned to 69 kV, continuous fluoroscopy, a 0.5 mm thick copperplate was attached to the X-ray detector, and 2 perspex plates with a thickness of 1 cm were placed between beam and detector. The beam current was adjusted to vary the image quality, as indicated for each picture. The images were recorded on 1 June 2004 at Philips Medical Systems, Best.*

	Hessian		*Meijering*		*Added m4*	
Image	*SBR*	*HC*	*SBR*	*HC*	*SBR*	*HC*
EP1	6.9 (3)	0.96 (3)	5.6 (3)	0.96 (2.75)	5.7 (3.25)	0.97 (2.75)
EP2	7.8 (4)	0.89 (4)	6.0 (3.75)	0.88 (3)	5.6 (4.25)	0.89 (3.5)
EP3	5.6 (3.5)	0.77 (4)	4.6 (3.25)	0.75 (3.5)	4.7 (3.5)	0.76 (4.5)
EP4	6.0 (3.25)	0.82 (5)	4.7 (3.25)	0.80 (3.75)	4.8 (3.5)	0.81 (4.25)
EP5	5.1 (4)	0.86 (3.75)	4.0 (3.75)	0.86 (3.5)	4.1 (4)	0.88 (3.75)
EP6	5,9 (2.75)	0.87 (2.75)	4.7 (2.75)	0.86 (2.75)	4.9 (2.75)	0.86 (2.75)
Mean	6.2 (3.4)	0.86 (3.8)	4.9 (3.3)	0.85 (3.2)	5.0 (3.5)	0.86 (3.6)

Table A.1: *Measured optimal values for SBR and HC, and the corresponding local scales (between brackets), in the 6 test images.*

	Hessian				*Added m4*			
Image	*SBR*		*HC*		*SBR*		*HC*	
EP1 1.5 mAs	8.4	9.7 (4)	0.98	0.99 (11)	6.0	8.8 (4)	0.99	0.99 (6)
EP1 0.2 mAs	3.0	3.3 (4)	0.61	0.66 (17)	2.1	3.0 (19)	0.55	0.68 (17)
EP2 1.5 mAs	8.0	10.4 (19)	0.89	0.96 (17)	5.3	8.6 (20)	0.9	0.95 (18)
EP2 0.2 mAs	4.6	6.0 (22)	0.71	0.77 (20)	3.1	5.2 (22)	0.73	0.78 (19)
EP3	7.3	9.0 (20)	0.78	0.97 (21)	4.9	8.0 (22)	0.78	0.95 (22)
EP4	8.0	8.5 (2)	0.80	0.81 (9)	5.0	7.0 (4)	0.80	0.81 (3)
EP5	5.1	6.0 (19)	0.82	0.92 (12)	3.4	5.2 (20)	0.85	0.90 (15)
EP6	7.9	8.9 (3)	0.88	0.92 (16)	5.4	7.5 (4)	0.90	0.92 (15)

Table A.2: *Effect of tensor voting on the 6 test images. For every measure, the number in the first column indicates the values* before *tensor voting. The next column shows the optimal value of the measure* after *tensor voting, and the corresponding context scales (between brackets).*

A.2 Evaluation of extraction results

Method	Noise	%fail	%frag	%tip	%tip+ext	%entire
Hessian+TV	Low	20	6	80	62	35
	High	28	3	72	37	13
Hessian	Low	43	7	57	42	18
	High	57	10	43	10	3
Added m4+TV	Low	27	2	73	53	34
	High	38	5	62	30	9
Added m4	Low	37	6	63	39	14
	High	62	8	38	7	0

Table A.3: *Succes and failure rates on the test set consisting of clinical images. The column %fail shows the percentage of failures (i.e., score 0 or 1 in Table 6.2), the column %frag the percentage of catheters with score 1, %tip the success rate on tip detection (score 2, 3, or 4), %tip+ext is the number of successful extracted tips with extension (score 3 or 4), and %entire is the percentage of catheter extracted entirely correct (score 4).*

Hessian+TV

Dose	%fail	%frag	%tip	%tip+ext	%entire
0.2	56	24	44	16	0
0.5	36	24	64	44	16
1	16	4	84	68	52
2	20	12	80	60	40
4	12	4	88	68	60
Mean	28	14	72	51	34

Hessian, no TV

Dose	%fail	%frag	%tip	%tip+ext	%entire
0.2	60	16	40	8	0
0.5	28	16	72	52	4
1	32	20	68	56	28
2	24	8	76	68	40
4	16	8	84	68	52
Mean	32	14	68	50	25

Added m4+TV

Dose	%fail	%frag	%tip	%tip+ext	%entire
0.2	60	20	40	24	0
0.5	36	16	64	48	20
1	12	4	88	64	48
2	20	12	80	64	48
4	8	0	92	72	68
Mean	27	10	73	54	37

Added m4, no TV

Dose	%fail	%frag	%tip	%tip+ext	%entire
0,2	76	20	24	8	0
0,5	48	32	52	48	20
1	20	8	80	56	36
2	24	12	76	60	44
4	12	8	88	72	64
Mean	36	16	64	49	33

Table A.4: *Success and failure rates on the test set consisting of phantom images. The column %fail shows the percentage of failures (i.e., score 0 and 1 in Table 6.2), the column %frag the percentage of catheters with score 1, %tip the success rate on tip detection (score 2, 3, or 4), %tip+ext is the number of successful extracted tips with extension (score 3 or 4), and %entire is the percentage of catheter extracted entirely correct (score 4).*

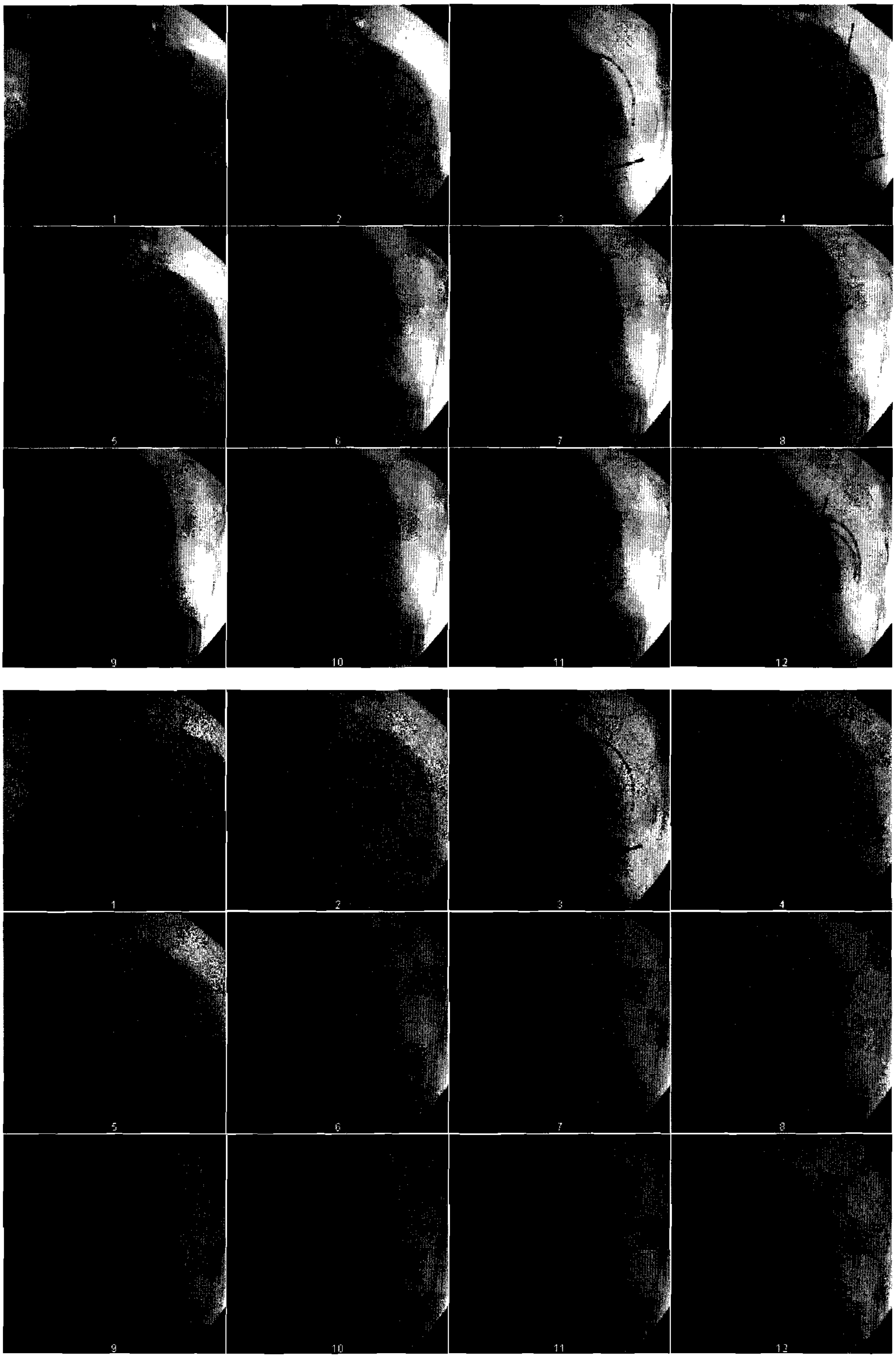


Figure A.3: *EP catheter extraction phantom test images, with highest dose (top) and lowest dose (bottom).*

Appendix B

Software

This appendix includes a short description of the software that was implemented during this project. In the appendix we refer to directories as they appear on the CD-ROM that is included with this report.

B.1 Getting started

The software was written in Mathematica and C++. The C++ functions are automatically called by Mathematica. Also, an extraction result viewer was program for ImageJ in Java. The CD-ROM contains the software in the directory `/software`.

The directory `/software/MmaPackages` contains a number of Mathematica packages, which are loaded into Mathematica by "`<<ErikFranken`'*package_name*'". The directory `ErikFranken` with the packages should be placed in one of the paths returned by Mathematica when evaluating `$AddOnsDirectory` or `$UserAddOnsDirectory` . This directory also includes a compiled version of the Mathlink program for Windows. It should be easy to compile the program on e.g. Linux or Macintosh, but this was not tested. The C++ source code is located in `/software/EpCathEx_cppsrc`.

B.2 Mathematica notebooks

The directory `/software/MmaNotebooks` contains notebooks that are useful for experimenting with the software. The notebooks are divided into the following subdirectories

- `EPCathextract/` contains notebooks that demonstrate the catheter extraction.
- `Report/` contains notebooks that were used to generate pictures for the report.
- `Misc/` contains various other notebooks.

B.3 Mathematica packages

This section contains a short overview of the packages. A short explanation of each function can be found by evaluating ?*function_name* in Mathematica.

For clarity, most function names have a prefix of a few characters, indicating the kind of function. A list of all functions of a class can be requested by ?*prefix** in Mathematica. The classes are:

- `Img*` general image processing functions.
- `LD*` functions related to local feature detection.
- `STV*` functions for steerable tensor voting
- `TV*` functions for normal tensor voting

- `CE*` functions for high-level catheter extraction (and visualization of the results)

Alternatively, one can find the functions of each package by opening the notebook file of the package.

ErikFranken‘MiscImageRoutines‘

This package contains useful miscellaneous image processing functions.

ErikFranken‘LocalFeatureDetect‘

This package contains functions for local feature detection (ridges, blobs, end-points) all based on Gaussian derivatives and steerable filters. Designed for use on (EP) X-ray images, but in principle useful for other images as well.

ErikFranken‘SteerableTensorVoting‘

This package contains functions for steerable tensor voting.

ErikFranken‘EPCathExMathlink‘

This package launches the MathLink program `EPCathEx.exe` and makes the functions and the documentation available.

ErikFranken‘EPCathExtract‘

This package contains functions for EP catheter extraction. Note, that the real extraction part is programmed in C++, in the MathLink program `EPCathEx.exe`, which should be used via the package `ErikFranken‘EPCathExMathlink‘`. This package only contains some additional functions, for instance for visualization of the results.

ErikFranken‘Measurements‘

This package contains functions for doing measurements in feature images. This was used for the quantitative analysis.

ErikFranken‘XrayMovies‘

This package contains functions for loading raw X-ray movies, and saving EP Catheter extraction results for a movie in a file format, suitable to be read by the ImageJ plugin (Section B.5).

ErikFranken‘TensorVoting‘

This package contains functions for tensor voting. Note, that it is recommended to use the implementation in `ErikFranken‘SteerableTensorVoting‘`. That tensor voting implementation better exploits the steerability of the kernel.

This package only contains some additional functions for tensor voting, for instance for visualization of the results. The real tensor voting functions to be used with this package are programmed in C++, and are included in the MathLink program `EPCathEx.exe`.

ErikFranken‘OrientationBundles‘

This package contains functions to work with orientation bundles.

B.4 C++ code

The C++ code was documented using *Doxygen* (see http://www.doxygen.org/), open `/Software/EpCathEx_cppsrc_doc/html/index.html` to browse through the documentation. Table B.1 shows a list with global descriptions of all C++ files.

File(s)	*Description*
`basics.h`	Basic inline functions and constants that are used in various other places.
`distancetransform.cpp`	Contains function for Euclidean distance transform of an image.
`EPcatheter.cpp`	Contains class to represent an EP catheter (path with corresponding electrodes), and functionality to create a EP catheter tip path from a detected group of electrodes.
`EPCathEx_tm.cpp`	File that is automatically generated by MathLink tool **`mprep`** from `EPCathEx.tm` (which contains function definitions for Mathematica).
`EPcathextract.cpp`	Contains class that links together all high-level extraction functionality in `EPelectrodes.cpp`, `EPcatheter.cpp`, `pathextract.cpp`, and `pathgrouping.cpp`.
`EPelectrodes.cpp`	Contains functions for extracting electrodes of EP catheters.
`extractsettings.cpp`	Contains a class that manages all settings for the high-level extraction part.
`fastmarcher.cpp` *	Contains class that contains Fast Marching functionality.
`featuremaps.cpp`	Contains classes to hold 2D maps of integers, doubles, and 2D vectors.
`image-io.cpp`	Contains functions to store pixel maps in PGM file format.
`imagefeatures.cpp`	Contains classes for representing grouping primitives (image features) of an image: points, paths, and end-points.
`logger.cpp`	Opens file stream `logger` for logging events.
`main.cpp`	Main file of the `EPCathEx` Mathlink program.
`mma_func.cpp`	Contains some functionality to make the use of MathLink somewhat more convenient.
`mmai_extractor.cpp`	Contains Mathlink interfacing functions for EP catheter extraction functions.
`mmai_misc.cpp`	Contains Mathlink interfacing functions for miscellaneous image processing functions.
`mmai_voting.cpp`	Contains Mathlink interfacing functions for tensor voting functions.
`morphblobextract.cpp` *	Contains function for blob extraction using a morphological operation, as designed by Philips.
`morphimgproc.cpp`	Contains standard morphological image processing functions.
`pathextract.cpp`	Contains functions for extracting paths, splitting paths, and some related helpful functions.
`pathgrouping.cpp`	Contains functions to group paths into longer paths (the so-called super paths).
`sparsify.cpp`	Contains functions to sparsify image data, for instance extraction of local maxima and thinning.
`steerfilter.cpp`	Contains function to steer a filter with $m = 0$, $\|m\| = 2$, and $\|m\| = 4$ components.
`tensorvoting.cpp`	Contains tensor voting functions.
`vect2d.cpp`	Contains classes to hold 2D vectors and 2D integer coordinates.

Table B.1: *Short description of all C++ files of the project. All* `.cpp` *files have a corresponding* `.h` *file (except for* `main.cpp`*). (*) indicates that the functionality in that file is not used in the EP catheter detection algorithm as proposed in this report.*

B.5 ImageJ plugin

To easily view extraction results on an EP image sequence, the package `ErikFranken'XrayMovies'` contains functions to save the extraction results of an image sequence in a file. To view the results, the ImageJ (see http://rsb.info.nih.gov/ij/) plugin `show_EPextractresult.java` (in directory `/software/ImageJplugin`) can be used. Figure B.1 shows a screen dump. The buttons below the image are added by the plugin. This enables the user to open other result files, to start playing the sequence, and to enable/disable visibility of the extraction result overlay.

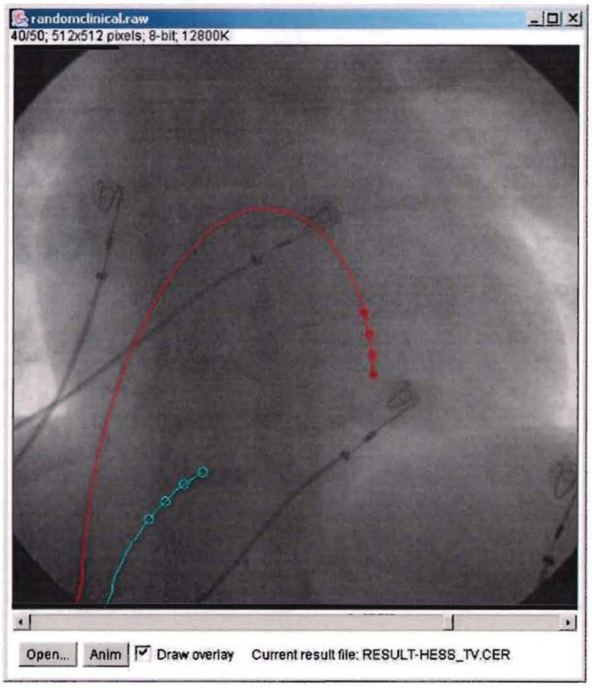


Figure B.1: *Screen dump of ImageJ plugin for viewing results.*

Appendix C

Other Applications

In this report, we focussed on the extraction of one very specific class of objects, i.e. EP catheters. There are many other medical image analysis problems that amount to extraction of elongated structures in noisy images. In this appendix, we show the potential use of our techniques for two other applications: detection of *guide wires* and detection of *blood vessels in mammography.*

C.1 Detection of guide wires

The detection of end-points of a guide wire is another problem of interest for Philips Medical Systems [13, 6, 28, 4]. A guide wire appears in the X-ray image as an elongated structure that is thinner than an EP catheter. The last part of the guide wire is more visible due to a special coating. For clinical practice, this *tip* is the most important part to detect. Especially accurate localization of the end-points of the tip is important.

C.1.1 Method

We will try to detect guide-wires using the algorithms for detecting EP catheters described in this thesis, with a few small modifications. We will only mention these modifications in this section.

As local filter, we use the Hessian. The value for local scale σ_{local} should be smaller than for EP catheters because guide-wires are thinner. We choose $\sigma_{\text{local}} = 2$. We use the same tensor voting scheme, but with smaller context scales, because the observation is that guide-wires can exhibit slightly larger curvature than EP catheters do. We choose $\sigma_{\text{ctx}} = 10$, $\sigma_{\text{ctx2}} = 7$.

The blob detector is omitted. Instead, we use an end-point detector to detect the end-points of the guide wire tip.

C.1.2 End-point detection

At an end-point of a line structure, the gradient in the direction tangent to the line structure is generally large, and also the ridgeness value will be large. Therefore, the absolute value of the inner product of the context enhanced ridgeness vector and the gradient vector, i.e.

$$E(x,y) = R(x,y) \left| \begin{pmatrix} \cos\Psi(x,y) \\ \sin\Psi(x,y) \end{pmatrix} \cdot \begin{pmatrix} L_\sigma^{(1,0)}(x,y) \\ L_\sigma^{(0,1)}(x,y) \end{pmatrix} \right|, \qquad \text{(C.1)}$$

is a measure for end-pointness. R and Ψ are the context enhanced ridgeness resp orientation (the output of Chapter 4), and $L_\sigma^{(1,0)}$ denotes the first order Gaussian derivative in x-direction at scale σ of the (background equalized) image. As scale for the end-point detection $\sigma = 4$ is chosen, which is twice the scale used for the Hessian filter.

Instead of the blobness, we supply the end-pointness to our high-level extraction algorithm. We let the EP catheter electrode detection algorithm search for groups of two end-points with an intermediate distance that is specific for a guide wire tip. We allow maximal 2 end-point candidates per path, because more than 2 end-points do not make sense. To fulfill this requirement, we only keep the two end-point candidates on a path with the largest distance in-between. Table C.1 shows the modified parameter values for feature image generation and high-level extraction, with their new meaning.

Name	*Value*	*New meaning*
σ_{local}	2	Local scale of guide wire
n	3	Voting field width
σ_{ctx}	10	Scale of voting field for first TV step
σ_{ctx2}	5	Scale of voting field for second TV step
`n_BlobExtract`	20	Number of end-point candidates to extract
`d_MinInterElectrode`	70	Minimum distance between the two end-points
`d_MaxInterElectrode`	150	Maximum distance between the two end-points
`ca_MaxElectrAngleChange`	0.0	Min. cosine of angle between end-points (must be permissive)
`n_MaxElectrodes`	2	Number of end-points of a guide wire tip (is always 2)
`n_MinElectrodes`	2	Number of end-points of a guide wire tip (is always 2)

Table C.1: *Overview of modified parameters for the detection of guide-wires. See Table 6.1 (page 98) and Table 5.2 (page 90) for the other parameters.*

C.1.3 Results

The guide wire extraction was tested on a small number of images (7) with guide-wires, with and without added multiplicative Poisson noise. Sometimes the extraction of the tip worked fine, see for example Figure C.1 and Figure C.2. However, in a lot of images the extraction went wrong. Often the wrong structure is selected. Extraction of the entire guide wire was only successful in one image. This is because the rest of the guide wire is often hardly visible, especially in the images with added noise, because in these images the rest of the guide wire seems not visible at all. Tensor voting leads to better tip extraction results for the images with added noise. For images without added noise, tensor voting does not lead to better tip extraction results, but it gives better results for extraction of the entire guide wire. Figure C.3 shows some more examples of guide wire extraction.

C.1.4 Conclusions

The method as proposed here is not yet feasible for guide wire detection. However, we observed that also in this application, tensor voting seems to lead to better results. Therefore, we conclude that the techniques could have added value for this application, but a more thorough investigation is needed. The algorithm should be adapted for this specific application, the parameters should be better optimized (for instance, now we just "guessed" the scale parameters), and it should be tested more extensively. The results will also improve if the algorithm only searches in a smaller region of interest in the image.

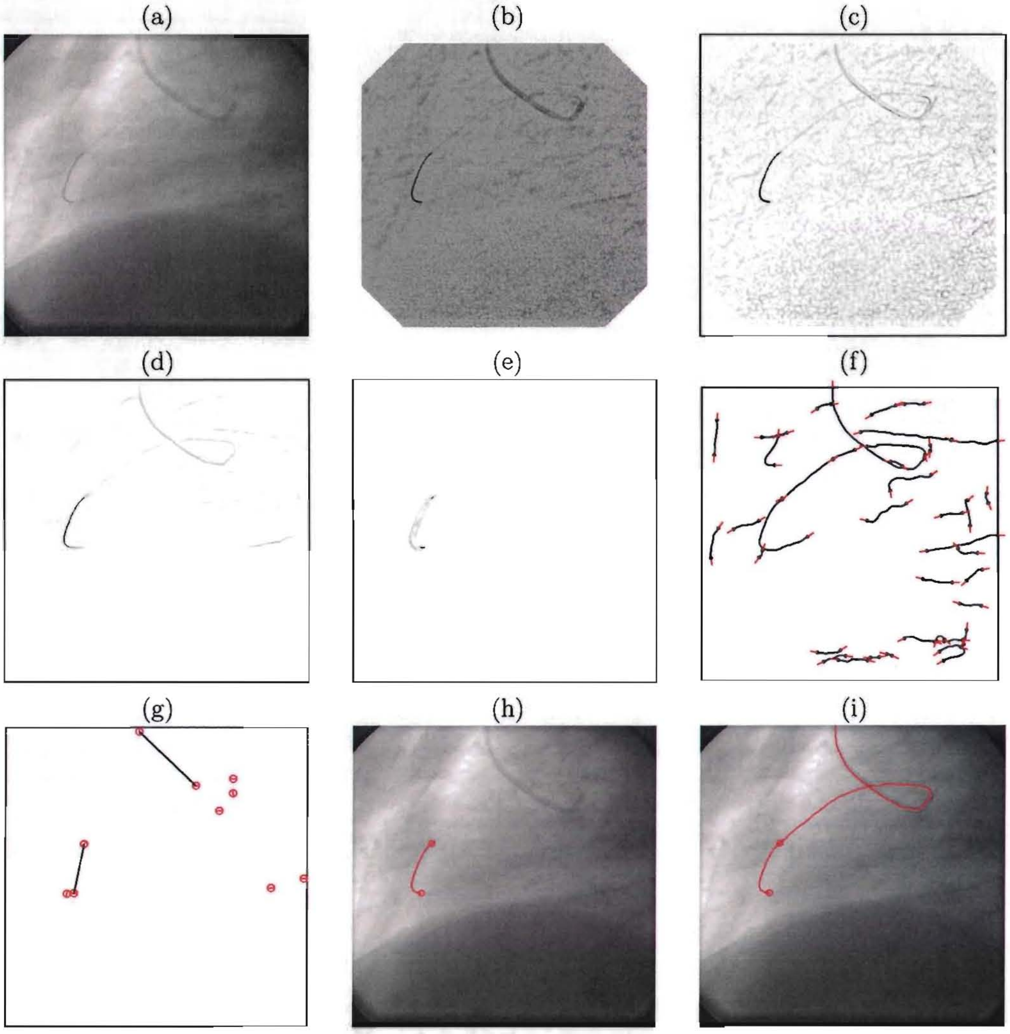


Figure C.1: *Example of step by step guide wire extraction. (a) Original image. (b) Contrast equalized image. (c) Local ridgeness. (d) Context enhanced ridgeness. (e) End-pointness. (f) Extracted paths. (g) End-point candidate graph. (h) Extracted guide wire tip. (i) Extracted guide wire.*

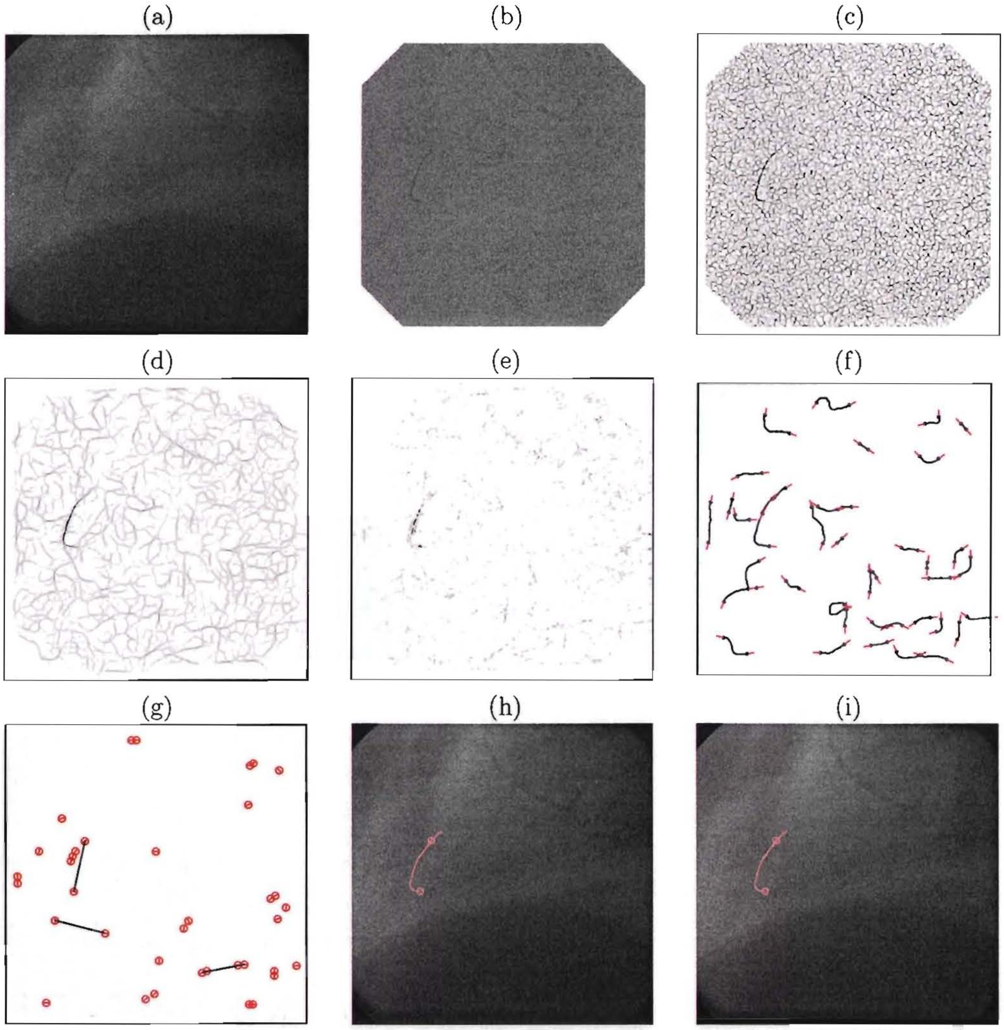


Figure C.2: *Example of step by step guide wire extraction, on the same image as Figure C.1 but additional noise added. (a) Original image. (b) Contrast equalized image. (c) Local ridgeness. (d) Context enhanced ridgeness. (e) End-pointness. (f) Extracted paths. (g) End-point candidate graph. (h) Extracted guide wire tip. (i) Extracted guide wire (failed).*

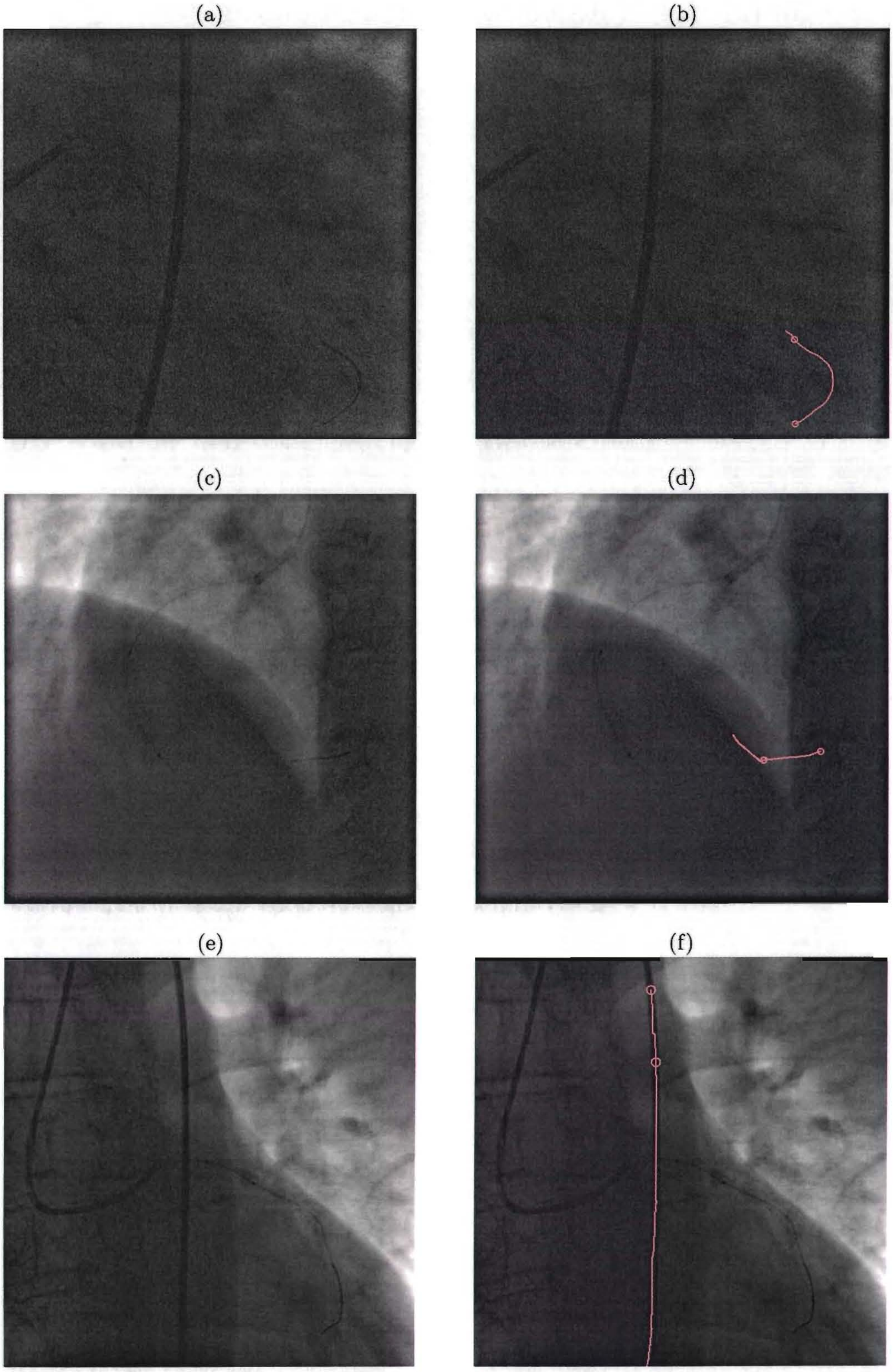


Figure C.3: *Examples of guide wire extraction. (a)-(b) Image with added noise. The tip detection is correct. (c)-(d) The tip is correct, the extension is wrong. (e)-(f) The guide wire is totally missed.*

C.2 Blood vessel detection in mammography

Mammography is a specific type of imaging that uses a low-dose X-ray system for examination of the breasts. Algorithms exist for automatic detection of breast cancer, which appear in mammography images as star-shaped structures. The existing algorithms, however, sometimes give false alarms at locations where blood vessels appear. Therefore, it is also of interest to detect the blood vessels in order to *exclude* these structures. This application is of interest for a project at Biomedical Image Analysis, TU/e, in collaboration with the university of Nijmegen.

We use the same approach as for detection of EP catheters, but we omit the contrast equalization step. We only extract paths and do not apply further high-level grouping, because this type of images is too different. We also do not know exactly which elongated structures in the image are the actual blood vessels. This should be discussed with the medical personnel. Table C.2 shows the modified parameters for mammography. Note, that these parameters are currently guessed.

Name	*Value*	*New meaning*
σ_{local}	2.4	Local scale of blood vessels
n	4	Voting field width
σ_{ctx}	8	Scale of voting field for first TV step
σ_{ctx2}	6	Scale of voting field for second TV step
`n_PathExtract`	100	Number of most salient paths to extract

Table C.2: *Overview of modified parameters for mammography. See Table 6.1 (98) and Table 5.2 (90) for the other parameters.*

Figure C.4 shows an example of the application of our techniques on a mammography image. It can be observed that tensor voting again simplifies the extraction of lines structures (compare Figure C.4d-e). At first sight, the result does not look bad, but a more detailed feasibility study is required.

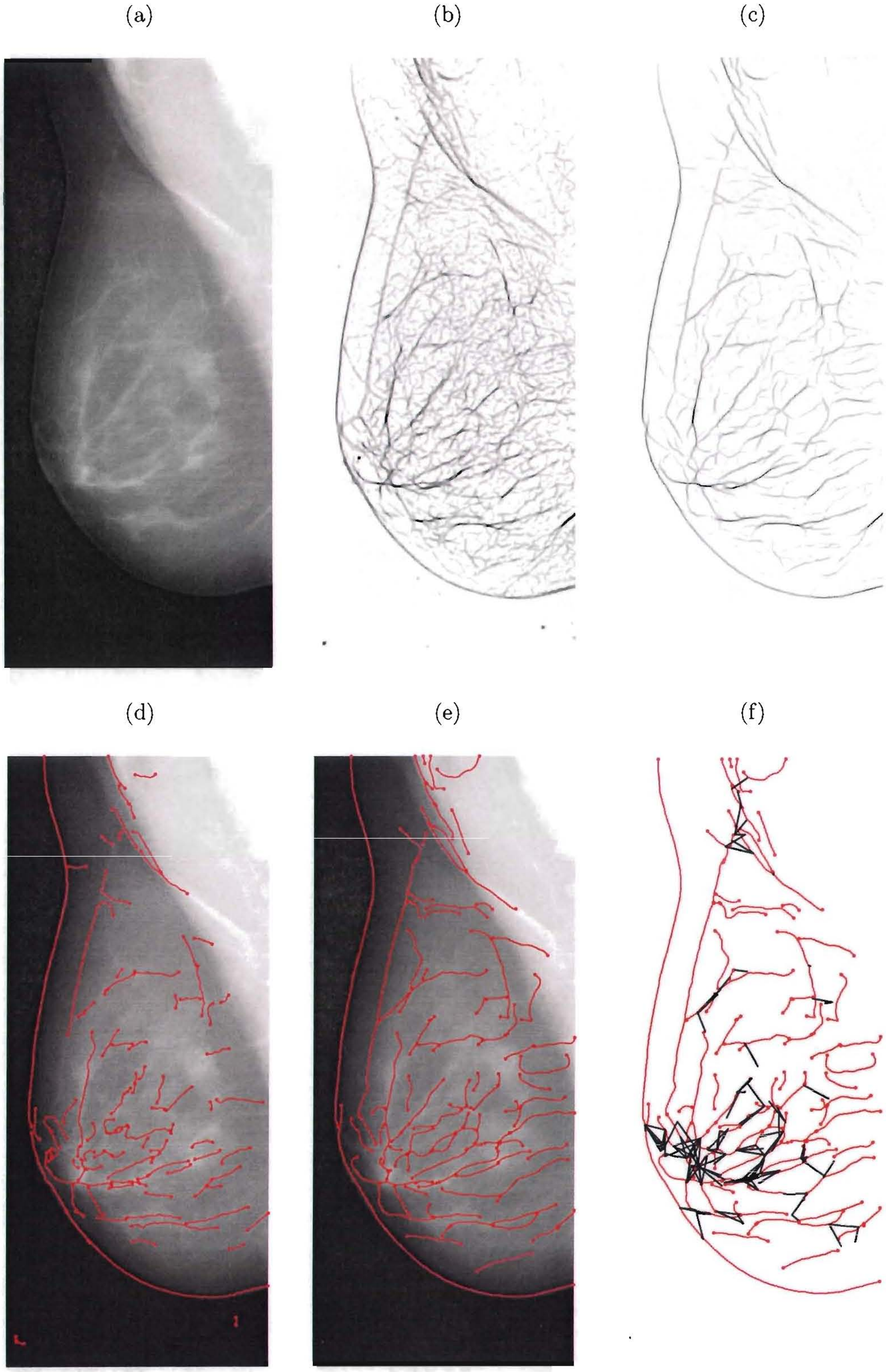


Figure C.4: *Example of blood vessel extraction in a mammography image. (a) Original image. (b) Local ridgeness. (c) Context enhanced ridgeness. (d) Extraction of 100 most salient paths from local ridgeness image. (e) Extraction of 100 most salient paths from context enhanced ridgeness image. Note, that these paths are smoother and longer, and thus probably better for further processing. (f) Path graph of paths in (e), where black lines indicate possible connections between paths. This information can be useful for further extraction of the blood vessels.*

Bibliography

[1] Heart rhythm society. website, 2004. URL: http://www.hrspatients.org/.

[2] N. Akhiezer and I. Glazman. *Theorie of Linear Operators in Hilbert Space*, volume 1. Frederick Ungar, New York, 1961.

[3] A. Amir and M. Lindenbaum. A generic grouping algorithm and its quantitative analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(2), feb 1998.

[4] S. Baert. *Guide wire tracking and reconstruction in interventional radiology.* PhD thesis, Image Sciences Institute, Utrecht, 2003.

[5] W. H. Bosking, Y. Zhang, B. Schofield, and D. Fitzpatrick. Orientation selectivity and the arrangement of horizontal connections in tree shrew striate cortex. *The Journal of Neuroscience*, 17(6):2112–2127, Mar. 1997.

[6] J. Breitenstein and R. Florent. Guide-wire extraction in x-ray fluoroscopy. Technical Report C 99-705, Philips Research, Jan. 1999.

[7] H. Breu, J. Gil, D. Kirkpatrick, and M. Werman. Linear time euclidean distance transform algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):529–533, may 1995.

[8] A. Davies, S. Kengyelics, A. Cowen, J. Moore, C. Pepper, and C. Cowan. Dose reduction in electrophysiology fluoroscopy. Technical report, Leeds X-ray Imaging Reserach, The University of Leeds, UK, 2004.

[9] R. Duits, M. van Almsick, M. Duits, E. Franken, and L. Florack. Image processing via shift-twist invariant operations on orientation bundle functions. In *7th International Conference on Pattern Recognition and Image Analysis (PRIA-7-2004)*, 2004.

[10] J. Feldman. Perceptual grouping by selection of a logically minimal model. *International Journal of Computer Vision*, 55(1):5–25, 2003.

[11] S. Fischer, P. Bayerl, H. Neumann, G. Cristóball, and R. Redondo. Are iterations and curvature useful for tensor voting? In T. Pajdla and J. Matas, editors, *Computer Vision - ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic*, number 3, pages 158–169, May 2004.

[12] L. Florack. *Image Structure.* Computational Imaging and Vision. Kluwer Academic Publishers, 1997.

[13] R. Florent and L. Goubet. Guide-wire extraction: Algorithmic overview and software implementation. Technical Report C 2000-741, Philips Research, June 2000.

[14] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991.

[15] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

[16] M. Hamermesh. *Group theory and its application to physical problems.* Addison-Wesley, 1962.

[17] T. Hansen. *A Neural Model of Early Vision: Contrast, Contours, Corners and Surfaces.* PhD thesis, University of Ulm, 2002.

[18] F. Heitger and R. von der Heydt. A computational model of neural contour processing: Figure-ground segregation and illusory contours. In *Proc. 4th Int. Conf. Computer Vision*, pages 32–40. Washington D.C.: IEEE Computer Society Press, 1993.

[19] D. H. Hubel. *Eye, Brain and Vision.* Scientific American Library, New York, 1988.

[20] M. Jacob and M. Unser. Design of steerable filters for feature detection using canny-like criteria. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):1007–1019, Aug. 2004.

[21] A. C. Jalba, M. H. Wilkinson, and J. B. Roerdink. Cpm: A deformable model for shape recovery and segmentation based on charged particles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10), Oct. 2004.

[22] S. N. Kalitzin, B. M. ter Haar Romeny, and M. A. Viergever. Invertible orientation bundles on 2d scalar images. In B. M. ter Haar Romeny, L. Florack, J. Koenderink, and M. Viergever, editors, *Scale Space Theory in Computer Vision*, pages 77–88, 1997.

[23] E. R. Kandel, J. H. Schwartz, and T. M. Jessell. *Principles of Neural Science.* Prentice-Hall, London, third edition, 1991.

[24] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1987.

[25] M. J. Kern. *The Cardiac Catheterization Handbook.* Mosby, fourth edition, 2003.

[26] D. Kunz and T. Aach. Relevant X-ray physics for noise reducation in fluoroscopy imaging. Technical Report 1129/95, Philips Forschungslaboratorien, Aachen, Germany, 1995.

[27] D. B. Kynor, A. J. Dietz, E. M. Friets, J. N. Peterson, U. C. Bergstrom, J. K. Triedman, and P. E. Hammer. Visualization of cardiac wavefronts using data fusion. In M. Sonka, editor, *Medical imaging 2002: Image Processing. Proceedings of the SPIE*, volume 4684, pages 1186–1194. SPIE-Int. Soc. Opt. Eng, 2002.

[28] P. Le Nézet. Tensor voting : 2 case studies. Technical Report S 2001-1072, Philips Recherche France, July 2001.

[29] T. Lindeberg. On Scale Selection for Differential Operators. In *Proc. 8th Scandinavian Conference on Image Analysis*, pages 857–866, Tromso, Norway, May 1993.

[30] A. Massad, M. Babós, and B. Mertsching. Application of the tensor voting technique for perceptual grouping to grey-level images. In L. van Gool, editor, *Pattern Recognition. 24th DAGM Symposium*, volume 2449, 2002.

[31] A. Massad and B. Mertsching. Quantitative comparison of tensorial image descriptions for the application to perceptual grouping by the tensor voting technique. In *The 3rd IASTED Intl. Conf. on Visualization, Imaging, and Image Processing (VIIP 2003), Benalmdena, Spain*, 2003.

[32] G. Medioni, M.-S. Lee, and C.-K. Tang. *A Computational Framework for Segmentation and Grouping.* Elsevier, 2000.

[33] E. Meijering, M. Jacob, J. F. Sarria, and M. Unser. A novel approach to neurite tracing in fluorescence microscopy images. In M. Hamza, editor, *Proceedings of the Fifth IASTED International Conference on Signal and Image Processing*, pages 491–495. ACTA Press, Calgary, 2003.

[34] S. Mollus, J. Weese, and H. Braess. Concepts for CNR imaging. Technical Report 1703/2004, Philips Forschungslaboratorien, Aachen, Germany, Jan. 2004.

[35] M. Sonka. *Handbook of Medical Imaging*, volume 2. SPIE, Belligham, 2001.

[36] C.-K. Tang and G. Medioni. Curvature-augmented tensor voting for shape inference from noisy 3d data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(6):858–864, June 2002.

[37] B. M. ter Haar Romeny. *Front-end vision vision and multi-scale image analysis : multi-scale computer vision theory and applications, written in mathematica.* Kluwer Academic Publishers, Oct. 2003.

[38] W.-S. Tong, C.-K. Tang, and G. Medioni. First order tensor voting, and application to 3-D scale analysis. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference*, volume 1, pages 175–182, 2001.

[39] W.-S. Tong, C.-K. Tang, P. Mordohai, and G. Medioni. First order augmentation to tensor voting for boundary inference and multiscale analysis in 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):594–611, May 2004.